User's Manual

32

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

SH7214 Group, SH7216 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine family/SH7216 Series

Renesas Electronics
www.renesas.com

Rev.4.00 Jun 2013

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 SH7214 and SH7216 Groups. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Data Sheet	Overview of hardware and electrical characteristics	—	—
User′s Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	SH7214 Group, SH7216 Group User′s Manual: Hardware	This user′s manual
User′s Manual: Software	Detailed descriptions of the CPU and instruction set	SH-2A, SH2A-FPU Software Manual	REJ09B0051
Application Note	Examples of applications and sample programs	The latest versions are available from our web site.	
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.		

2. Description of Numbers and Symbols

Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

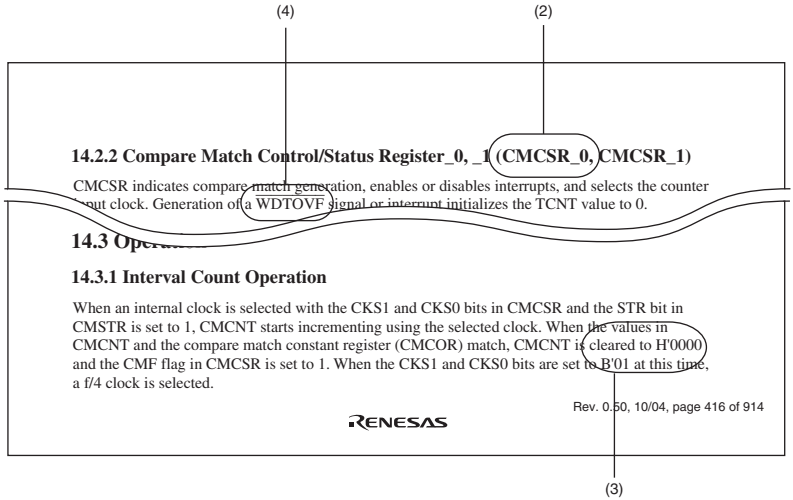
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.
[Examples] Binary: B'11 or 11
Hexadecimal: H'EFA0 or 0xEFA0
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.
[Example] $\overline{\text{WDTOVF}}$

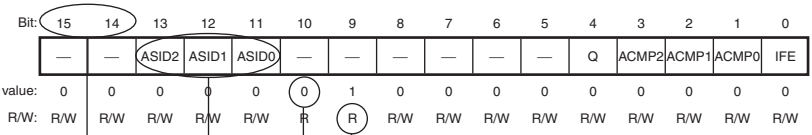


Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

3. 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.

[Bit Chart]



[Table of Bits]

(1) (2) (3) (4) (5)

Bit	Bit Name	Initial Value	R/W	Description
15	–	0	R	Reserved
14	–	0	R	These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10	–	0	R	Reserved This bit is always read as 0.
9	–	1	R	Reserved This bit is always read as 1.
	–	0		

Note: The bit names and sentences in the above figure are examples, and have nothing to do with the contents of this manual.

(1) Bit
Indicates the bit number or numbers.
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.

(2) Bit name
Indicates the name of the bit or bit field.
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).
A reserved bit is indicated by "–".
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.

(3) Initial value
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.
0: The initial value is 0
1: The initial value is 1
–: The initial value is undefined

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.
The notation is as follows:

R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

(5) Description
Describes the function of the bit or field and specifies the values for writing.

4. Description of Abbreviations

The abbreviations used in this manual are listed below.

- Abbreviations specific to this product

Abbreviation	Description
BSC	Bus controller
CPG	Clock pulse generator
DTC	Data transfer controller
INTC	Interrupt controller
SCI	Serial communication interface
WDT	Watchdog timer

- Abbreviations other than those listed above

Abbreviation	Description
ACIA	Asynchronous communications interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	No connection
PLL	Phase-locked loop
PWM	Pulse width modulation
SFR	Special function register
SIM	Subscriber Identity Module
UART	Universal asynchronous receiver/transmitter
VCO	Voltage-controlled oscillator

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

Contents

Section 1 Overview 1
1.1 Features 1
1.2 Block Diagram 10
1.3 Pin Assignment 11
1.4 Pin Functions 13

Section 2 CPU 23
2.1 Data Formats 23
2.2 Register Descriptions 24
2.2.1 General Registers 24
2.2.2 Control Registers 25
2.2.3 System Registers 27
2.2.4 Floating-Point Registers 28
2.2.5 Floating-Point System Registers 29
2.2.6 Register Bank 32
2.2.7 Initial Values of Registers 32
2.3 Data Formats 33
2.3.1 Data Format in Registers 33
2.3.2 Data Formats in Memory 33
2.3.3 Immediate Data Format 34
2.4 Instruction Features 35
2.4.1 RISC-Type Instruction Set 35
2.4.2 Addressing Modes 39
2.4.3 Instruction Format 44
2.5 Instruction Set 48
2.5.1 Instruction Set by Classification 48
2.5.2 Data Transfer Instructions 55
2.5.3 Arithmetic Operation Instructions 59
2.5.4 Logic Operation Instructions 62
2.5.5 Shift Instructions 63
2.5.6 Branch Instructions 64
2.5.7 System Control Instructions 66
2.5.8 Floating-Point Operation Instructions 68
2.5.9 FPU-Related CPU Instructions 70
2.5.10 Bit Manipulation Instructions 70
2.6 Processing States 72

Section 3 MCU Operating Modes 75
3.1 Selection of Operating Modes 75
3.2 Input/Output Pins 76
3.3 Operating Modes 76
3.3.1 Mode 0 (MCU Extension Mode 0) 76
3.3.2 Mode 1 (MCU Extension Mode 1) 76
3.3.3 Mode 2 (MCU Extension Mode 2) 76
3.3.4 Mode 3 (Single Chip Mode) 76
3.4 Address Map 77
3.5 Initial State in This LSI 80
3.6 Note on Changing Operating Mode 80

Section 4 Clock Pulse Generator (CPG) 81
4.1 Features 81
4.2 Input/Output Pins 85
4.3 Clock Operating Modes 86
4.4 Register Descriptions 90
4.4.1 Frequency Control Register (FRQCR) 90
4.4.2 MTU2S Clock Frequency Control Register (MCLKCR) 93
4.4.3 AD Clock Frequency Control Register (ACLKCR) 94
4.4.4 Oscillation Stop Detection Control Register (OSCCR) 95
4.5 Changing the Frequency 96
4.6 Oscillator 97
4.6.1 Connecting Crystal Resonator 97
4.6.2 External Clock Input Method 98
4.7 Oscillation Stop Detection 99
4.8 USB Operating Clock (48 MHz) 100
4.8.1 Connecting a Ceramic Resonator 100
4.8.2 Input of an External 48-MHz Clock Signal 101
4.8.3 Handling of pins when a Ceramic Resonator is not Connected (the Internal CPG is Selected or the USB is Not in Use) 102
4.9 Notes on Board Design 103
4.9.1 Note on Using an External Crystal Resonator 103

Section 5 Exception Handling 105
5.1 Overview 105
5.1.1 Types of Exception Handling and Priority 105
5.1.2 Exception Handling Operations 107
5.1.3 Exception Handling Vector Table 109
5.2 Resets 111
5.2.1 Types of Reset 111

5.2.2 Power-On Reset ... 112
5.2.3 Manual Reset ... 113
5.3 Address Errors ... 115
5.3.1 Address Error Sources ... 115
5.3.2 Address Error Exception Handling ... 116
5.4 Register Bank Errors ... 117
5.4.1 Register Bank Error Sources ... 117
5.4.2 Register Bank Error Exception Handling ... 117
5.5 Interrupts ... 118
5.5.1 Interrupt Sources ... 118
5.5.2 Interrupt Priority Level ... 119
5.5.3 Interrupt Exception Handling ... 120
5.6 Exceptions Triggered by Instructions ... 121
5.6.1 Types of Exceptions Triggered by Instructions ... 121
5.6.2 Trap Instructions ... 122
5.6.3 Slot Illegal Instructions ... 122
5.6.4 General Illegal Instructions ... 123
5.6.5 Integer Division Instructions ... 123
5.6.6 Floating Point Operation Instruction ... 124
5.7 When Exception Sources Are Not Accepted ... 125
5.8 Stack Status after Exception Handling Ends ... 126
5.9 Usage Notes ... 128
5.9.1 Value of Stack Pointer (SP) ... 128
5.9.2 Value of Vector Base Register (VBR) ... 128
5.9.3 Address Errors Caused by Stacking of Address Error Exception Handling ... 128
5.9.4 Note When Changing Interrupt Mask Level (IMASK) of Status Register (SR) in CPU ... 128

Section 6 Interrupt Controller (INTC) ... 129
6.1 Features ... 129
6.2 Input/Output Pins ... 131
6.3 Register Descriptions ... 132
6.3.1 Interrupt Priority Registers 01, 02, 05 to 19 (IPR01, IPR02, IPR05 to IPR19) 133
6.3.2 Interrupt Control Register 0 (ICR0) ... 135
6.3.3 Interrupt Control Register 1 (ICR1) ... 136
6.3.4 IRQ Interrupt Request Register (IRQRR) ... 137
6.3.5 Bank Control Register (IBCR) ... 139
6.3.6 Bank Number Register (IBNR) ... 140
6.3.7 USB-DTC Transfer Interrupt Request Register (USDTENDRR) ... 141
6.4 Interrupt Sources ... 143
6.4.1 NMI Interrupt ... 143
6.4.2 User Break Interrupt ... 143

6.4.3 H-UDI Interrupt ... 143
6.4.4 IRQ Interrupts ... 144
6.4.5 Memory Error Interrupt ... 144
6.4.6 On-Chip Peripheral Module Interrupts ... 145
6.5 Interrupt Exception Handling Vector Table and Priority ... 146
6.6 Operation ... 155
6.6.1 Interrupt Operation Sequence ... 155
6.6.2 Stack after Interrupt Exception Handling ... 158
6.7 Interrupt Response Time ... 159
6.8 Register Banks ... 165
6.8.1 Banked Register and Input/Output of Banks ... 166
6.8.2 Bank Save and Restore Operations ... 166
6.8.3 Save and Restore Operations after Saving to All Banks ... 168
6.8.4 Register Bank Exception ... 169
6.8.5 Register Bank Error Exception Handling ... 169
6.9 Data Transfer with Interrupt Request Signals ... 170
6.9.1 Handling Interrupt Request Signals as DTC Activating Sources and CPU Interrupt Sources but Not as DMAC Activating Sources ... 172
6.9.2 Handling Interrupt Request Signals as DMAC Activating Sources but Not as CPU Interrupt Sources ... 172
6.9.3 Handling Interrupt Request Signals as DTC Activating Sources but Not as CPU Interrupt Sources or DMAC Activating Sources ... 172
6.9.4 Handling Interrupt Request Signals as CPU Interrupt Sources but Not as DTC Activating Sources or DMAC Activating Sources ... 173
6.10 Usage Notes ... 173
6.10.1 Timing to Clear an Interrupt Source ... 173
6.10.2 In Case the NMI Pin is not in Use ... 173
6.10.3 Negate Timing of $\overline{\text{IRQOUT}}$ ... 173
6.10.4 Notes on Canceling Software Standby Mode with an IRQx Interrupt Request ... 174

Section 7 User Break Controller (UBC) ... 175
7.1 Features ... 175
7.2 Input/Output Pin ... 177
7.3 Register Descriptions ... 178
7.3.1 Break Address Register_0 (BAR_0) ... 179
7.3.2 Break Address Mask Register_0 (BAMR_0) ... 180
7.3.3 Break Bus Cycle Register_0 (BBR_0) ... 181
7.3.4 Break Address Register_1 (BAR_1) ... 183
7.3.5 Break Address Mask Register_1 (BAMR_1) ... 184
7.3.6 Break Bus Cycle Register_1 (BBR_1) ... 185
7.3.7 Break Address Register_2 (BAR_2) ... 187

7.3.8 Break Address Mask Register_2 (BAMR_2) ... 188
7.3.9 Break Bus Cycle Register_2 (BBR_2) ... 189
7.3.10 Break Address Register_3 (BAR_3) ... 191
7.3.11 Break Address Mask Register_3 (BAMR_3) ... 192
7.3.12 Break Bus Cycle Register_3 (BBR_3) ... 193
7.3.13 Break Control Register (BRCR) ... 195
7.4 Operation ... 199
7.4.1 Flow of the User Break Operation ... 199
7.4.2 Break on Instruction Fetch Cycle ... 200
7.4.3 Break on Data Access Cycle ... 201
7.4.4 Value of Saved Program Counter ... 202
7.4.5 Usage Examples ... 203
7.5 Interrupt Source ... 205
7.6 Usage Notes ... 206

Section 8 Data Transfer Controller (DTC) ... 207
8.1 Features ... 207
8.2 Register Descriptions ... 209
8.2.1 DTC Mode Register A (MRA) ... 210
8.2.2 DTC Mode Register B (MRB) ... 211
8.2.3 DTC Source Address Register (SAR) ... 212
8.2.4 DTC Destination Address Register (DAR) ... 213
8.2.5 DTC Transfer Count Register A (CRA) ... 214
8.2.6 DTC Transfer Count Register B (CRB) ... 215
8.2.7 DTC Enable Registers A to E (DTCERA to DTCERE) ... 216
8.2.8 DTC Control Register (DTCCR) ... 217
8.2.9 DTC Vector Base Register (DTCVBR) ... 218
8.2.10 Bus Function Extending Register (BSCEHR) ... 219
8.3 Activation Sources ... 219
8.4 Location of Transfer Information and DTC Vector Table ... 219
8.5 Operation ... 224
8.5.1 Transfer Information Read Skip Function ... 229
8.5.2 Transfer Information Write-Back Skip Function ... 230
8.5.3 Normal Transfer Mode ... 230
8.5.4 Repeat Transfer Mode ... 231
8.5.5 Block Transfer Mode ... 233
8.5.6 Chain Transfer ... 234
8.5.7 Operation Timing ... 236
8.5.8 Number of DTC Execution Cycles ... 239
8.5.9 DTC Bus Release Timing ... 242
8.5.10 DTC Activation Priority Order ... 244
8.6 DTC Activation by Interrupt ... 246

8.7 Examples of Use of the DTC ... 247
8.7.1 Normal Transfer Mode ... 247
8.7.2 Chain Transfer when Transfer Counter = 0 ... 248
8.8 Interrupt Sources ... 249
8.9 Usage Notes ... 249
8.9.1 Module Standby Mode Setting ... 249
8.9.2 On-Chip RAM ... 250
8.9.3 DTCE Bit Setting ... 250
8.9.4 Chain Transfer ... 250
8.9.5 Transfer Information Start Address, Source Address, and Destination Address ... 250
8.9.6 Access to DTC Registers through DTC ... 250
8.9.7 Notes on IRQ Interrupt as DTC Activation Source ... 250
8.9.8 Note on SCI or SCIF as DTC Activation Sources ... 251
8.9.9 Clearing Interrupt Source Flag ... 251
8.9.10 Conflict between NMI Interrupt and DTC Activation ... 251
8.9.11 Note on USB as DTC Activation Sources ... 251
8.9.12 Operation when a DTC Activation Request has been Cancelled ... 251
8.9.13 Note on Writing to DTCER ... 251

Section 9 Bus State Controller (BSC) ... 253
9.1 Features ... 253
9.2 Input/Output Pins ... 256
9.3 Area Overview ... 257
9.3.1 Address Map ... 257
9.3.2 Setting Operating Modes ... 260
9.4 Register Descriptions ... 261
9.4.1 Common Control Register (CMNCR) ... 262
9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7) ... 265
9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7) ... 270
9.4.4 SDRAM Control Register (SDCR) ... 299
9.4.5 Refresh Timer Control/Status Register (RTCSR) ... 303
9.4.6 Refresh Timer Counter (RTCNT) ... 305
9.4.7 Refresh Time Constant Register (RTCOR) ... 306
9.4.8 Bus Function Extending Register (BSCEHR) ... 307
9.5 Operation ... 310
9.5.1 Endian/Access Size and Data Alignment ... 310
9.5.2 Normal Space Interface ... 314
9.5.3 Access Wait Control ... 319
9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion ... 321
9.5.5 MPX-I/O Interface ... 322
9.5.6 SDRAM Interface ... 327

9.5.7 Burst ROM (Clock Asynchronous) Interface ... 369
9.5.8 SRAM Interface with Byte Selection ... 371
9.5.9 Burst ROM (Clock Synchronous) Interface ... 376
9.5.10 Wait between Access Cycles ... 377
9.5.11 Bus Arbitration ... 385
9.5.12 Others ... 387
9.6 Interrupt Source ... 395
9.7 Usage Note ... 396
9.7.1 Note on Connection of External LSI Circuits such as SDRAMs and ASICs ... 396

Section 10 Direct Memory Access Controller (DMAC) ... 397
10.1 Features ... 397
10.2 Input/Output Pins ... 399
10.3 Register Descriptions ... 400
10.3.1 DMA Source Address Registers (SAR) ... 405
10.3.2 DMA Destination Address Registers (DAR) ... 406
10.3.3 DMA Transfer Count Registers (DMATCR) ... 407
10.3.4 DMA Channel Control Registers (CHCR) ... 408
10.3.5 DMA Reload Source Address Registers (RSAR) ... 416
10.3.6 DMA Reload Destination Address Registers (RDAR) ... 417
10.3.7 DMA Reload Transfer Count Registers (RDMATCR) ... 418
10.3.8 DMA Operation Register (DMAOR) ... 419
10.3.9 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3) ... 423
10.4 Operation ... 425
10.4.1 Transfer Flow ... 425
10.4.2 DMA Transfer Requests ... 427
10.4.3 Channel Priority ... 431
10.4.4 DMA Transfer Types ... 434
10.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing ... 443
10.5 Interrupt Sources ... 447
10.5.1 Interrupt Sources and Priority Order ... 447
10.6 Usage Notes ... 449
10.6.1 Setting of the Half-End Flag and the Half-End Interrupt ... 449
10.6.2 Timing of DACK and TEND Outputs ... 449
10.6.3 CHCR Setting ... 449
10.6.4 Note on Activation of Multiple Channels ... 449
10.6.5 Note on Transfer Request Input ... 449
10.6.6 Conflict between NMI Interrupt and DMAC Activation ... 450
10.6.7 Number of On-Chip RAM Access Cycles from DMAC ... 450

Section 11 Multi-Function Timer Pulse Unit 2 (MTU2) 451
11.1 Features 451
11.2 Input/Output Pins 457
11.3 Register Descriptions 458
11.3.1 Timer Control Register (TCR) 462
11.3.2 Timer Mode Register (TMDR) 466
11.3.3 Timer I/O Control Register (TIOR) 469
11.3.4 Timer Compare Match Clear Register (TCNTCMPCLR) 488
11.3.5 Timer Interrupt Enable Register (TIER) 489
11.3.6 Timer Status Register (TSR) 494
11.3.7 Timer Buffer Operation Transfer Mode Register (TBTM) 501
11.3.8 Timer Input Capture Control Register (TICCR) 503
11.3.9 Timer Synchronous Clear Register S (TSYCRS) 504
11.3.10 Timer A/D Converter Start Request Control Register (TADCR) 506
11.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) 509
11.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) 509
11.3.13 Timer Counter (TCNT) 510
11.3.14 Timer General Register (TGR) 510
11.3.15 Timer Start Register (TSTR) 511
11.3.16 Timer Synchronous Register (TSYR) 513
11.3.17 Timer Counter Synchronous Start Register (TCSYSTR) 515
11.3.18 Timer Read/Write Enable Register (TRWER) 518
11.3.19 Timer Output Master Enable Register (TOER) 519
11.3.20 Timer Output Control Register 1 (TOCR1) 520
11.3.21 Timer Output Control Register 2 (TOCR2) 523
11.3.22 Timer Output Level Buffer Register (TOLBR) 526
11.3.23 Timer Gate Control Register (TGCR) 527
11.3.24 Timer Subcounter (TCNTS) 529
11.3.25 Timer Dead Time Data Register (TDDR) 530
11.3.26 Timer Cycle Data Register (TCDR) 530
11.3.27 Timer Cycle Buffer Register (TCBR) 531
11.3.28 Timer Interrupt Skipping Set Register (TITCR) 531
11.3.29 Timer Interrupt Skipping Counter (TITCNT) 533
11.3.30 Timer Buffer Transfer Set Register (TBTER) 534
11.3.31 Timer Dead Time Enable Register (TDER) 536
11.3.32 Timer Waveform Control Register (TWCR) 537
11.3.33 Bus Master Interface 539
11.4 Operation 540
11.4.1 Basic Functions 540
11.4.2 Synchronous Operation 546

11.4.3 Buffer Operation ... 548
11.4.4 Cascaded Operation ... 552
11.4.5 PWM Modes ... 557
11.4.6 Phase Counting Mode ... 562
11.4.7 Reset-Synchronized PWM Mode ... 569
11.4.8 Complementary PWM Mode ... 572
11.4.9 A/D Converter Start Request Delaying Function ... 614
11.4.10 MTU2-MTU2S Synchronous Operation ... 619
11.4.11 External Pulse Width Measurement ... 625
11.4.12 Dead Time Compensation ... 626
11.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operation ... 629
11.5 Interrupt Sources ... 630
11.5.1 Interrupt Sources and Priorities ... 630
11.5.2 DMAC and DTC Activation ... 632
11.5.3 A/D Converter Activation ... 633
11.6 Operation Timing ... 635
11.6.1 Input/Output Timing ... 635
11.6.2 Interrupt Signal Timing ... 642
11.7 Usage Notes ... 648
11.7.1 Module Standby Mode Setting ... 648
11.7.2 Input Clock Restrictions ... 648
11.7.3 Caution on Period Setting ... 649
11.7.4 Contention between TCNT Write and Clear Operations ... 649
11.7.5 Contention between TCNT Write and Increment Operations ... 650
11.7.6 Contention between TGR Write and Compare Match ... 651
11.7.7 Contention between Buffer Register Write and Compare Match ... 652
11.7.8 Contention between Buffer Register Write and TCNT Clear ... 653
11.7.9 Contention between TGR Read and Input Capture ... 654
11.7.10 Contention between TGR Write and Input Capture ... 655
11.7.11 Contention between Buffer Register Write and Input Capture ... 656
11.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection ... 656
11.7.13 Counter Value during Complementary PWM Mode Stop ... 658
11.7.14 Buffer Operation Setting in Complementary PWM Mode ... 658
11.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag ... 659
11.7.16 Overflow Flags in Reset Synchronous PWM Mode ... 660
11.7.17 Contention between Overflow/Underflow and Counter Clearing ... 661
11.7.18 Contention between TCNT Write and Overflow/Underflow ... 662
11.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode ... 662
11.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode ... 663
11.7.21 Interrupts in Module Standby Mode ... 663

11.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection 663
11.7.23 Note on Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode 664
11.8 MTU2 Output Pin Initialization 666
11.8.1 Operating Modes 666
11.8.2 Reset Start Operation 666
11.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc. 667
11.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. 668

Section 12 Multi-Function Timer Pulse Unit 2S (MTU2S) 699
12.1 Input/Output Pins 702
12.2 Register Descriptions 703

Section 13 Port Output Enable 2 (POE2) 707
13.1 Features 707
13.2 Input/Output Pins 709
13.3 Register Descriptions 711
13.3.1 Input Level Control/Status Register 1 (ICSR1) 712
13.3.2 Output Level Control/Status Register 1 (OCSR1) 716
13.3.3 Input Level Control/Status Register 2 (ICSR2) 717
13.3.4 Output Level Control/Status Register 2 (OCSR2) 718
13.3.5 Input Level Control/Status Register 3 (ICSR3) 720
13.3.6 Software Port Output Enable Register (SPOER) 722
13.3.7 Port Output Enable Control Register 1 (POECR1) 723
13.3.8 Port Output Enable Control Register 2 (POECR2) 725
13.4 Operation 731
13.4.1 Input Level Detection Operation 732
13.4.2 Output-Level Compare Operation 734
13.4.3 Release from High-Impedance State 734
13.5 Interrupts 735
13.6 Usage Notes 736
13.6.1 Pins States when the Watchdog Timer has Issued a Power-on Reset 736
13.6.2 Input Pins 736

Section 14 Compare Match Timer (CMT) 737
14.1 Features 737
14.2 Register Descriptions 738
14.2.1 Compare Match Timer Start Register (CMSTR) 739
14.2.2 Compare Match Timer Control/Status Register (CMCSR) 740
14.2.3 Compare Match Counter (CMCNT) 742

14.2.4 Compare Match Constant Register (CMCOR) ... 742
14.3 Operation ... 743
14.3.1 Interval Count Operation ... 743
14.3.2 CMCNT Count Timing ... 743
14.4 Interrupts ... 744
14.4.1 Interrupt Sources and DTC/DMAC Transfer Requests ... 744
14.4.2 Timing of Compare Match Flag Setting ... 745
14.4.3 Timing of Compare Match Flag Clearing ... 745
14.5 Usage Notes ... 746
14.5.1 Conflict between Write and Compare-Match Processes of CMCNT ... 746
14.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT ... 747
14.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT ... 748
14.5.4 Compare Match between CMCNT and CMCOR ... 748

Section 15 Watchdog Timer (WDT) ... 749
15.1 Features ... 749
15.2 Input/Output Pin ... 750
15.3 Register Descriptions ... 751
15.3.1 Watchdog Timer Counter (WTCNT) ... 751
15.3.2 Watchdog Timer Control/Status Register (WTCSR) ... 752
15.3.3 Watchdog Reset Control/Status Register (WRCSR) ... 754
15.3.4 Notes on Register Access ... 755
15.4 WDT Usage ... 757
15.4.1 Canceling Software Standby Mode ... 757
15.4.2 Using Watchdog Timer Mode ... 757
15.4.3 Using Interval Timer Mode ... 759
15.5 Interrupt Sources ... 760
15.6 Usage Notes ... 761
15.6.1 Timer Variation ... 761
15.6.2 Prohibition against Setting H'FF to WTCNT ... 761
15.6.3 Interval Timer Overflow Flag ... 761
15.6.4 System Reset by $\overline{\text{WDTOVF}}$ Signal ... 762
15.6.5 Manual Reset in Watchdog Timer Mode ... 762
15.6.6 Connection of the $\overline{\text{WDTOVF}}$ Pin ... 762

Section 16 Serial Communication Interface (SCI) ... 763
16.1 Features ... 763
16.2 Input/Output Pins ... 765
16.3 Register Descriptions ... 766
16.3.1 Receive Shift Register (SCRSR) ... 767
16.3.2 Receive Data Register (SCRDR) ... 767

16.3.3 Transmit Shift Register (SCTSR) ... 768
16.3.4 Transmit Data Register (SCTDR) ... 768
16.3.5 Serial Mode Register (SCSMR) ... 768
16.3.6 Serial Control Register (SCSCR) ... 772
16.3.7 Serial Status Register (SCSSR) ... 775
16.3.8 Serial Port Register (SCSPTR) ... 781
16.3.9 Serial Direction Control Register (SCSDCR) ... 783
16.3.10 Bit Rate Register (SCBRR) ... 784
16.4 Operation ... 796
16.4.1 Overview ... 796
16.4.2 Operation in Asynchronous Mode ... 798
16.4.3 Clock Synchronous Mode ... 809
16.4.4 Multiprocessor Communication Function ... 818
16.4.5 Multiprocessor Serial Data Transmission ... 820
16.4.6 Multiprocessor Serial Data Reception ... 821
16.5 SCI Interrupt Sources and DTC ... 824
16.6 Serial Port Register (SCSPTR) and SCI Pins ... 825
16.7 Usage Notes ... 827
16.7.1 SCTDR Writing and TDRE Flag ... 827
16.7.2 Multiple Receive Error Occurrence ... 827
16.7.3 Break Detection and Processing ... 828
16.7.4 Sending a Break Signal ... 828
16.7.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) ... 828
16.7.6 Note on Using DTC ... 830
16.7.7 Note on Using External Clock in Clock Synchronous Mode ... 830
16.7.8 Module Standby Mode Setting ... 830

Section 17 Serial Communication Interface with FIFO (SCIF) ... 831
17.1 Features ... 831
17.2 Input/Output Pins ... 833
17.3 Register Descriptions ... 833
17.3.1 Receive Shift Register (SCRSR) ... 834
17.3.2 Receive FIFO Data Register (SCFRDR) ... 834
17.3.3 Transmit Shift Register (SCTSR) ... 835
17.3.4 Transmit FIFO Data Register (SCFTDR) ... 835
17.3.5 Serial Mode Register (SCSMR) ... 836
17.3.6 Serial Control Register (SCSCR) ... 839
17.3.7 Serial Status Register (SCFSR) ... 843
17.3.8 Bit Rate Register (SCBRR) ... 851
17.3.9 FIFO Control Register (SCFCR) ... 863
17.3.10 FIFO Data Count Register (SCFDR) ... 865
17.3.11 Serial Port Register (SCSPTR) ... 866

17.3.12 Line Status Register (SCLSR) ... 868
17.3.13 Serial Extended Mode Register (SCSEMR) ... 869
17.4 Operation ... 870
17.4.1 Overview ... 870
17.4.2 Operation in Asynchronous Mode ... 872
17.4.3 Operation in Clocked Synchronous Mode ... 882
17.5 SCIF Interrupts ... 891
17.6 Usage Notes ... 892
17.6.1 SCFTDR Writing and TDFE Flag ... 892
17.6.2 SCFRDR Reading and RDF Flag ... 892
17.6.3 Restriction on DMAC and DTC Usage ... 893
17.6.4 Break Detection and Processing ... 893
17.6.5 Sending a Break Signal ... 893
17.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) ... 894
17.6.7 FER Flag and PER Flag of Serial Status Register (SCFSR) ... 895

Section 18 Renesas Serial Peripheral Interface (RSPI) ... 897
18.1 Features ... 897
18.1.1 Internal Block Diagram ... 899
18.2 Input/Output Pins ... 901
18.3 Register Descriptions ... 902
18.3.1 RSPI Control Register (SPCR) ... 903
18.3.2 RSPI Slave Select Polarity Register (SSLP) ... 906
18.3.3 RSPI Pin Control Register (SPPCR) ... 907
18.3.4 RSPI Status Register (SPSR) ... 908
18.3.5 RSPI Data Register (SPDR) ... 913
18.3.6 RSPI Sequence Control Register (SPSCR) ... 915
18.3.7 RSPI Sequence Status Register (SPSSR) ... 916
18.3.8 RSPI Bit Rate Register (SPBR) ... 918
18.3.9 RSPI Data Control Register (SPDCR) ... 919
18.3.10 RSPI Clock Delay Register (SPCKD) ... 923
18.3.11 SPI Slave Select Negation Delay Register (SSLND) ... 924
18.3.12 RSPI Next-Access Delay Register (SPND) ... 925
18.3.13 RSPI Command Register (SPCMD) ... 926
18.4 Operation ... 931
18.4.1 Overview of RSPI Operations ... 931
18.4.2 Controlling RSPI Pins ... 933
18.4.3 RSPI System Configuration Example ... 935
18.4.4 Transfer Format ... 944
18.4.5 Data Format ... 947
18.4.6 Transmit Buffer Empty/Receive Buffer Full Flags ... 953
18.4.7 Error Detection ... 955

18.4.8 Initializing RSPI ... 960
18.4.9 SPI Operation ... 961
18.4.10 Clock Synchronous Operation ... 973
18.4.11 Error Processing ... 980
18.4.12 Loopback Mode ... 982
18.4.13 Interrupt Request ... 983
18.5 Usage Notes ... 984
18.5.1 DTC Block Transfer ... 984
18.5.2 DMAC Burst Transfer ... 984
18.5.3 Reading Receive Data ... 984
18.5.4 DTC/DMAC and Mode Fault Error ... 984
18.5.5 Usage of the RSPI Output Pins as Open Drain Outputs ... 984

Section 19 I²C Bus Interface 3 (IIC3) ... 985
19.1 Features ... 985
19.2 Input/Output Pins ... 987
19.3 Register Descriptions ... 988
19.3.1 I²C Bus Control Register 1 (ICCR1) ... 989
19.3.2 I²C Bus Control Register 2 (ICCR2) ... 992
19.3.3 I²C Bus Mode Register (ICMR) ... 994
19.3.4 I²C Bus Interrupt Enable Register (ICIER) ... 996
19.3.5 I²C Bus Status Register (ICSR) ... 998
19.3.6 Slave Address Register (SAR) ... 1001
19.3.7 I²C Bus Transmit Data Register (ICDRT) ... 1001
19.3.8 I²C Bus Receive Data Register (ICDRR) ... 1002
19.3.9 I²C Bus Shift Register (ICDRS) ... 1002
19.3.10 NF2CYC Register (NF2CYC) ... 1003
19.4 Operation ... 1004
19.4.1 I²C Bus Format ... 1004
19.4.2 Master Transmit Operation ... 1005
19.4.3 Master Receive Operation ... 1007
19.4.4 Slave Transmit Operation ... 1009
19.4.5 Slave Receive Operation ... 1012
19.4.6 Clocked Synchronous Serial Format ... 1014
19.4.7 Noise Filter ... 1017
19.4.8 Using the IICRST Bit to Reset I²C Bus Interface 3 ... 1018
19.4.9 Example of Use ... 1019
19.5 Interrupt Requests ... 1023
19.6 Data Transfer Using DTC ... 1024
19.7 Bit Synchronous Circuit ... 1025
19.8 Usage Notes ... 1027
19.8.1 Setting for Multi-Master Operation ... 1027

19.8.2 Note on Master Receive Mode 1027
19.8.3 Note on Setting ACKBT in Master Receive Mode 1027
19.8.4 Note on the States of Bits MST and TRN when Arbitration Is Lost 1028
19.8.5 Access to ICE and IICRST Bits during I²C Bus Operations 1028
19.8.6 Using the IICRST Bit to Initialize the Registers 1029
19.8.7 Operation of I²C Bus Interface 3 while ICE = 0 1029
19.8.8 Note on Master Transmit Mode 1029

Section 20 A/D Converter (ADC) 1031
20.1 Features 1031
20.2 Input/Output Pins 1033
20.3 Register Descriptions 1034
20.3.1 A/D Control Registers 0 and 1 (ADCR_0 and ADCR_1) 1035
20.3.2 A/D Status Registers 0 to 1 (ADSR_0 and ADSR_1) 1038
20.3.3 A/D Start Trigger Select Registers 0 and 1 (ADSTRGR_0 and ADSTRGR_1) 1039
20.3.4 A/D Analog Input Channel Select Registers 0 and 1 (ADANSR_0 and ADANSR_1) 1041
20.3.5 A/D Bypass Control Registers 0 and 1 (ADBYPSCR_0 and ADBYPSCR_1) 1042
20.3.6 A/D Data Registers 0 to 7 (ADDR0 to ADDR7) 1043
20.4 Operation 1044
20.4.1 Single-Cycle Scan Mode 1044
20.4.2 Continuous Scan Mode 1047
20.4.3 Input Sampling and A/D Conversion Time 1050
20.4.4 A/D Converter Activation by MTU2 and MTU2S 1052
20.4.5 External Trigger Input Timing 1052
20.4.6 Example of ADDR Auto-Clear Function 1053
20.5 Interrupt Sources and DMAC or DTC Transfer Requests 1055
20.6 Definitions of A/D Conversion Accuracy 1056
20.7 Usage Notes 1058
20.7.1 Analog Input Voltage Range 1058
20.7.2 Relationship between AVcc, AVss and VccQ, Vss 1058
20.7.3 Range of AVREF Pin Settings 1058
20.7.4 Notes on Board Design 1058
20.7.5 Notes on Noise Countermeasures 1059
20.7.6 Notes on Register Setting 1059
20.7.7 Permissible Signal Source Impedance 1060
20.7.8 Influences on Absolute Precision 1060
20.7.9 Notes when Two A/D Modules Run Simultaneously 1060

Section 21 Controller Area Network (RCAN-ET) 1063
21.1 Summary 1063
21.1.1 Overview 1063
21.1.2 Scope 1063
21.1.3 Audience 1063
21.1.4 References 1064
21.1.5 Features 1064
21.2 Architecture 1065
21.3 Programming Model - Overview 1067
21.3.1 Memory Map 1067
21.3.2 Mailbox Structure 1068
21.3.3 RCAN-ET Control Registers 1075
21.3.4 RCAN-ET Mailbox Registers 1095
21.4 Application Note 1106
21.4.1 Test Mode Settings 1106
21.4.2 Configuration of RCAN-ET 1107
21.4.3 Message Transmission Sequence 1113
21.4.4 Message Receive Sequence 1116
21.4.5 Reconfiguration of Mailbox 1118
21.5 Interrupt Sources 1120
21.6 DTC Interface 1121
21.7 DMAC Interface 1122
21.8 CAN Bus Interface 1123
21.9 Usage Notes 1124
21.9.1 Module Standby Mode 1124
21.9.2 Reset 1124
21.9.3 CAN Sleep Mode 1124
21.9.4 Register Access 1124
21.9.5 Interrupts 1125

Section 22 Pin Function Controller (PFC) 1127
22.1 Register Descriptions 1143
22.1.1 Port A I/O Registers H and L (PAIORH and PAIORL) 1145
22.1.2 Port A Control Registers H1 and H2, and L1 to L4 (PACRH1 and PACRH2, and PACRL1 to PACRL4) 1146
22.1.3 Port A Pull-Up MOS Control Registers H and L (PAPCRH and PAPCRL) .. 1158
22.1.4 Port B I/O Register L (PBIORL) 1160
22.1.5 Port B Control Registers L1 to L4 (PBCRL1 to PBCRL4) 1160
22.1.6 Port B Pull-Up MOS Control Register L (PBPCRL) 1169
22.1.7 Port C I/O Register L (PCIORL) 1170
22.1.8 Port C Control Registers L1 to L4 (PCCRL1 to PCCRL4) 1170
22.1.9 Port C Pull-Up MOS Control Register L (PCPCRL) 1179

22.1.10 Port D I/O Registers H and L (PDIORH and PDIORL) 1180
22.1.11 Port D Control Registers H1 to H4 and L1 to L4 (PDCRH1 to PDCRH4 and PDCRL1 to PDCRL4) 1181
22.1.12 Port D Pull-Up MOS Control Registers H and L (PDPCRH and PDPCRL) 1198
22.1.13 Port E I/O Register L (PEIORL) 1200
22.1.14 Port E Control Registers L1 to L4 (PECRL1 to PECRL4) 1201
22.1.15 Port E Pull-Up MOS Control Register L (PEPCRL) 1210
22.1.16 Large Current Port Control Register (HCPCR) 1211
22.1.17 IRQOUT Function Control Register (IFCR) 1213
22.1.18 DACK Output Timing Control Register (PDACKCR) 1214
22.2 Pull-Up MOS Control by Pin Function 1219
22.3 Usage Notes 1223

Section 23 I/O Ports 1225
23.1 Port A 1225
23.1.1 Register Descriptions 1226
23.1.2 Port A Data Registers H and L (PADRH and PADRL) 1226
23.1.3 Port A Port Registers H and L (PAPRH and PAPRL) 1228
23.2 Port B 1230
23.2.1 Register Descriptions 1230
23.2.2 Port B Data Register L PBDRL) 1231
23.2.3 Port B Port Register L (PBPRL) 1232
23.3 Port C 1233
23.3.1 Register Descriptions 1234
23.3.2 Port C Data Register L (PCDRL) 1234
23.3.3 Port C Port Register L (PCPRL) 1236
23.4 Port D 1237
23.4.1 Register Descriptions 1238
23.4.2 Port D Data Registers H and L (PDDRH and PDDRL) 1238
23.4.3 Port D Port Registers H and L (PDPRH and PDPRL) 1241
23.5 Port E 1243
23.5.1 Register Descriptions 1243
23.5.2 Port E Data Register L (PEDRL) 1244
23.5.3 Port E Port Register L (PEPRL) 1245
23.6 Port F 1246
23.6.1 Register Descriptions 1246
23.6.2 Port F Data Register L (PFDRL) 1247
23.7 Usage Notes 1248
23.7.1 Handling of Unused pins 1248

Section 24 USB Function Module (USB) 1249
24.1 Features 1249
24.2 Pin Configuration 1251
24.3 Register Descriptions 1252
24.3.1 USB Interrupt Flag Register 0 (USBIFR0) 1254
24.3.2 USB Interrupt Flag Register 1 (USBIFR1) 1255
24.3.3 USB Interrupt Flag Register 2 (USBIFR2) 1257
24.3.4 USB Interrupt Flag Register 3 (USBIFR3) 1258
24.3.5 USB Interrupt Flag Register 4 (USBIFR4) 1260
24.3.6 USB Interrupt Enable Register 0 (USBIER0) 1261
24.3.7 USB Interrupt Enable Register 1 (USBIER1) 1262
24.3.8 USB Interrupt Enable Register 2 (USBIER2) 1263
24.3.9 USB Interrupt Enable Register 3 (USBIER3) 1264
24.3.10 USB Interrupt Enable Register 4 (USBIER4) 1265
24.3.11 USB Interrupt Select Register 0 (USBISR0) 1266
24.3.12 USB Interrupt Select Register 1 (USBISR1) 1267
24.3.13 USB Interrupt Select Register 2 (USBISR2) 1268
24.3.14 USB Interrupt Select Register 3 (USBISR3) 1269
24.3.15 USB Interrupt Select Register 4 (USBISR4) 1270
24.3.16 USBEP0i Data Register (USBEPDR0i) 1271
24.3.17 USBEP0o Data Register (USBEPDR0o) 1271
24.3.18 USBEP0s Data Register (USBEPDR0s) 1272
24.3.19 USBEP1 Data Register (USBEPDR1) 1273
24.3.20 USBEP2 Data Register (USBEPDR2) 1273
24.3.21 USBEP3 Data Register (USBEPDR3) 1274
24.3.22 USBEP4 Data Register (USBEPDR4) 1274
24.3.23 USBEP5 Data Register (USBEPDR5) 1275
24.3.24 USBEP6 Data Register (USBEPDR6) 1275
24.3.25 USBEP7 Data Register (USBEPDR7) 1276
24.3.26 USBEP8 Data Register (USBEPDR8) 1276
24.3.27 USBEP9 Data Register (USBEPDR9) 1277
24.3.28 USBEP0o Receive Data Size Register (USBEPSZ0o) 1277
24.3.29 USBEP1 Receive Data Size Register (USBEPSZ1) 1278
24.3.30 USBEP4 Receive Data Size Register (USBEPSZ4) 1278
24.3.31 USBEP7 Receive Data Size Register (USBEPSZ7) 1279
24.3.32 USB Data Status Register 0 (USBDASTS0) 1279
24.3.33 USB Data Status Register 1 (USBDASTS1) 1280
24.3.34 USB Data Status Register 2 (USBDASTS2) 1281
24.3.35 USB Data Status Register 3 (USBDASTS3) 1282
24.3.36 USB Trigger Register 0 (USBTRG0) 1283
24.3.37 USB Trigger Register 1 (USBTRG1) 1284
24.3.38 USB Trigger Register 2 (USBTRG2) 1285

24.3.39 USB Trigger Register 3 (USBTRG3) ... 1286
24.3.40 USB FIFO Clear Register 0 (USBFCLR0) ... 1287
24.3.41 USB FIFO Clear Register 1 (USBFCLR1) ... 1288
24.3.42 USB FIFO Clear Register 2 (USBFCLR2) ... 1289
24.3.43 USB FIFO Clear Register 3 (USBFCLR3) ... 1290
24.3.44 USB Endpoint Stall Register 0 (USBEPSTL0) ... 1291
24.3.45 USB Endpoint Stall Register 1 (USBEPSTL1) ... 1292
24.3.46 USB Endpoint Stall Register 2 (USBEPSTL2) ... 1293
24.3.47 USB Endpoint Stall Register 3 (USBEPSTL3) ... 1294
24.3.48 USB Stall Status Register 1 (USBSTLSR1) ... 1296
24.3.49 USB Stall Status Register 2 (USBSTLSR2) ... 1298
24.3.50 USB Stall Status Register 3 (USBSTLSR3) ... 1300
24.3.51 USB DMA Transfer Setting Register (USBDMAR) ... 1302
24.3.52 USB Configuration Value Register (USBCVR) ... 1305
24.3.53 USB Control Register (USBCTLR) ... 1306
24.3.54 USB Endpoint Information Register (USBEPIR) ... 1307
24.3.55 USB Transceiver Test Register 0 (USBTRNTREG0) ... 1312
24.3.56 USB Transceiver Test Register 1 (USBTRNTREG1) ... 1314
24.4 Interrupt Sources ... 1316
24.5 Operation ... 1319
24.5.1 Initial Settings ... 1319
24.5.2 Cable Connection ... 1320
24.5.3 Cable Disconnection ... 1321
24.5.4 Control Transfer ... 1322
24.5.5 EP1/EP4/EP7 Bulk-OUT Transfer ... 1330
24.5.6 EP2/EP5/EP8 Bulk-IN Transfer ... 1332
24.5.7 EP3/EP6/EP9 Interrupt-IN Transfer ... 1334
24.6 Processing of USB Standard Commands and Class/Vendor Commands ... 1335
24.6.1 Processing of Commands Transmitted by Control Transfer ... 1335
24.7 Stall Operations ... 1336
24.7.1 Overview ... 1336
24.7.2 Forcible Stall by Application ... 1336
24.7.3 Automatic Stall by USB Function Module ... 1338
24.8 DMA Transfer ... 1339
24.8.1 Overview ... 1339
24.8.2 DMA Transfer for Endpoints 1 and 4 ... 1339
24.8.3 DMA Transfer for Endpoints 2 and 5 ... 1342
24.9 DTC Transfer ... 1346
24.9.1 DTC Transfer for Endpoints 1 and 4 ... 1346
24.9.2 DTC Transfer for Endpoints 2 and 5 ... 1350
24.10 Example of USB External Circuitry ... 1353
24.11 Usage Notes ... 1355

24.11.1 Receiving Setup Data ... 1355
24.11.2 Clearing FIFO ... 1355
24.11.3 Overreading or Overwriting Data Registers ... 1355
24.11.4 Assigning Interrupt Sources for EP0 ... 1356
24.11.5 Clearing FIFO when Setting DMAC/DTC Transfer ... 1356
24.11.6 Manual Reset for DMAC/DTC Transfer ... 1356
24.11.7 USB Clock ... 1356
24.11.8 Using TR Interrupt ... 1357
24.11.9 Handling of Unused USB Pins ... 1358

Section 25 Ethernet Controller (EtherC) (SH7216A, SH7214A, SH7216G, and SH7214G only) ... 1359
25.1 Features ... 1359
25.2 Input/Output Pins ... 1361
25.3 Register Descriptions ... 1362
25.3.1 EtherC Mode Register (ECMR) ... 1365
25.3.2 EtherC Status Register (ECSR) ... 1369
25.3.3 EtherC Interrupt Enable Register (ECSIPR) ... 1371
25.3.4 PHY Interface Register (PIR) ... 1373
25.3.5 MAC Address High Register (MAHR) ... 1374
25.3.6 MAC Address Low Register (MALR) ... 1375
25.3.7 Receive Frame Length Register (RFLR) ... 1376
25.3.8 PHY Status Register (PSR) ... 1377
25.3.9 Transmit Retry Over Counter Register (TROCR) ... 1378
25.3.10 Delayed Collision Detect Counter Register (CDCR) ... 1379
25.3.11 Lost Carrier Counter Register (LCCR) ... 1380
25.3.12 Carrier Not Detect Counter Register (CNDCR) ... 1381
25.3.13 CRC Error Frame Receive Counter Register (CEFCR) ... 1382
25.3.14 Frame Receive Error Counter Register (FRECR) ... 1383
25.3.15 Too-Short Frame Receive Counter Register (TSFRCR) ... 1384
25.3.16 Too-Long Frame Receive Counter Register (TLFRCR) ... 1385
25.3.17 Residual-Bit Frame Receive Counter Register (RFCR) ... 1386
25.3.18 Multicast Address Frame Receive Counter Register (MAFCR) ... 1387
25.3.19 IPG Register (IPGR) ... 1388
25.3.20 Automatic PAUSE Frame Register (APR) ... 1389
25.3.21 Manual PAUSE Frame Register (MPR) ... 1390
25.3.22 Automatic PAUSE Frame Retransmit Count Register (TPAUSER) ... 1391
25.3.23 Random Number Generation Counter Upper Limit Register (RDMLR) ... 1392
25.3.24 PAUSE Frame Receive Counter Register (RFCF) ... 1393
25.3.25 PAUSE Frame Retransmit Counter Register (TPAUSECR) ... 1394
25.3.26 Broadcast Frame Receive Count Register (BCFRR) ... 1395
25.4 Operation ... 1396

25.4.1 Transmission ... 1396
25.4.2 Reception ... 1399
25.4.3 MII Frame Timing ... 1401
25.4.4 Accessing MII Registers ... 1403
25.4.5 Magic Packet Detection ... 1406
25.4.6 Operation by IPG Setting ... 1407
25.4.7 Flow Control ... 1408
25.5 Connection to the PHY-LSI ... 1409
25.6 Usage Notes ... 1410

Section 26 Ethernet Controller Direct Memory Access Controller (E-DMAC) (SH7216A, SH7214A, SH7216G, and SH7214G only) ... 1411
26.1 Features ... 1411
26.2 Register Descriptions ... 1412
26.2.1 E-DMAC Mode Register (EDMR) ... 1414
26.2.2 E-DMAC Transmit Request Register (EDTRR) ... 1415
26.2.3 E-DMAC Receive Request Register (EDRRR) ... 1416
26.2.4 Transmit Descriptor List Start Address Register (TDLAR) ... 1417
26.2.5 Receive Descriptor List Start Address Register (RDLAR) ... 1418
26.2.6 EtherC/E-DMAC Status Register (EESR) ... 1419
26.2.7 EtherC/E-DMAC Status Interrupt Enable Register (EESIPR) ... 1424
26.2.8 Transmit/Receive Status Copy Enable Register (TRSCER) ... 1427
26.2.9 Receive Missed-Frame Counter Register (RMFCR) ... 1430
26.2.10 Transmit FIFO Threshold Register (TFTR) ... 1431
26.2.11 FIFO Depth Register (FDR) ... 1433
26.2.12 Receiving Method Control Register (RMCR) ... 1435
26.2.13 Transmit FIFO Underrun Counter Register (TFUCR) ... 1437
26.2.14 Receive FIFO Overflow Counter Register (RFOCR) ... 1438
26.2.15 Receive Buffer Write Address Register (RBWAR) ... 1439
26.2.16 Receive Descriptor Fetch Address Register (RDFAR) ... 1440
26.2.17 Transmit Buffer Read Address Register (TBRAR) ... 1441
26.2.18 Transmit Descriptor Fetch Address Register (TDFAR) ... 1442
26.2.19 Flow Control Start FIFO Threshold Setting Register (FCFTR) ... 1443
26.2.20 Transmit Interrupt Setting Register (TRIMD) ... 1445
26.2.21 Independent Output Signal Setting Register (IOSR) ... 1446
26.2.22 E-DMAC Operation Control Register (EDOCR) ... 1447
26.3 Operation ... 1449
26.3.1 Descriptor Lists and Data Buffers ... 1449
26.3.2 Transmission ... 1458
26.3.3 Reception ... 1460
26.3.4 Transmit/Receive Processing of Multi-Buffer Frame ... 1462
26.4 Usage Notes ... 1463

Section 27 Flash Memory (ROM) 1465
27.1 Features 1465
27.2 Input/Output Pins 1470
27.3 Register Descriptions 1471
27.3.1 Flash Pin Monitor Register (FPMON) 1472
27.3.2 Flash Mode Register (FMODR) 1473
27.3.3 Flash Access Status Register (FASTAT) 1474
27.3.4 Flash Access Error Interrupt Enable Register (FAEINT) 1476
27.3.5 ROM MAT Select Register (ROMMAT) 1477
27.3.6 FCU RAM Enable Register (FCURAME) 1478
27.3.7 Flash Status Register 0 (FSTATR0) 1479
27.3.8 Flash Status Register 1 (FSTATR1) 1483
27.3.9 Flash P/E Mode Entry Register (FENTRYR) 1484
27.3.10 Flash Protect Register (FPROTR) 1486
27.3.11 Flash Reset Register (FRESETR) 1487
27.3.12 FCU Command Register (FCMDR) 1488
27.3.13 FCU Processing Switch Register (FCPSR) 1489
27.3.14 Flash P/E Status Register (FPESTAT) 1490
27.3.15 ROM Cache Control Register (RCCR) 1491
27.3.16 Peripheral Clock Notification Register (PCKAR) 1492
27.4 Overview of ROM-Related Modes 1493
27.5 Boot Mode 1496
27.5.1 System Configuration 1496
27.5.2 State Transition in Boot Mode 1497
27.5.3 Automatic Adjustment of Bit Rate 1499
27.5.4 USB Boot Mode 1500
27.5.5 Inquiry/Selection Host Command Wait State 1504
27.5.6 Programming/Erasing Host Command Wait State 1522
27.6 User Program Mode 1534
27.6.1 FCU Command List 1534
27.6.2 Conditions for FCU Command Acceptance 1537
27.6.3 FCU Command Usage 1541
27.6.4 Suspending Operation 1560
27.7 User Boot Mode 1563
27.7.1 User Boot Mode Initiation 1563
27.7.2 User MAT Programming 1565
27.8 Programmer Mode 1566
27.9 Protection 1566
27.9.1 Hardware Protection 1566
27.9.2 Software Protection 1567
27.9.3 Error Protection 1568
27.10 Usage Notes 1571

27.10.1 Switching between User MAT and User Boot MAT 1571
27.10.2 State in which Interrupts are Ignored 1573
27.10.3 Programming-/Erasure-Suspended Area 1573
27.10.4 Compatibility with Programming/ Erasing Program of Conventional F-ZTAT SH Microcomputers 1573
27.10.5 FWE Pin State 1573
27.10.6 Reset during Programming or Erasure 1574
27.10.7 Suspension by Programming/Erasure Suspension 1574
27.10.8 Prohibition of Additional Programming 1575
27.10.9 Allocation of Interrupt Vectors during Programming and Erasure 1575
27.10.10 Items Prohibited during Programming and Erasure 1575
27.10.11 Abnormal Ending of Programming or Erasure 1575

Section 28 Data Flash (FLD) 1577
28.1 Features 1577
28.2 Input/Output Pins 1582
28.3 Register Descriptions 1582
28.3.1 Flash Mode Register (FMODR) 1584
28.3.2 Flash Access Status Register (FASTAT) 1585
28.3.3 Flash Access Error Interrupt Enable Register (FAEINT) 1588
28.3.4 FLD Read Enable Register 0 (EEPRE0) 1590
28.3.5 FLD Program/Erase Enable Register 0 (EEPWE0) 1591
28.3.6 Flash P/E Mode Entry Register (FENTRYR) 1592
28.3.7 FLD Blank Check Register (EEPBCCNT) 1594
28.3.8 FLD Blank Check Status Register (EEPBCSTAT) 1595
28.4 Overview of FLD-Related Modes 1596
28.5 Boot Mode 1598
28.5.1 Inquiry/Selection Host Commands 1598
28.5.2 Programming/Erasing Host Commands 1601
28.6 User Mode, User Program Mode, and User Boot Mode 1603
28.6.1 FCU Command List 1603
28.6.2 Conditions for FCU Command Acceptance 1605
28.6.3 FCU Command Usage 1609
28.7 Protection 1614
28.7.1 Hardware Protection 1614
28.7.2 Software Protection 1614
28.7.3 Error Protection 1615
28.8 Usage Notes 1617
28.8.1 Protection of Data MAT Immediately after a Reset 1617
28.8.2 State in which Interrupts are Ignored 1617
28.8.3 Programming-/Erasure-Suspended Area 1617

28.8.4 Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcontrollers ... 1617
28.8.5 Reset during Programming or Erasure ... 1618
28.8.6 Suspension by Programming/Erasure Suspension ... 1618
28.8.7 Prohibition of Additional Programming ... 1618
28.8.8 Program for Reading ... 1618
28.8.9 Items Prohibited during Programming and Erasure ... 1619
28.8.10 Abnormal Ending of Programming or Erasure ... 1619
28.8.11 Handling when Erasure or Programming is Stopped ... 1619

Section 29 On-Chip RAM ... 1621
29.1 Features ... 1621
29.2 Register Descriptions ... 1623
29.2.1 System Control Register 1 (SYSCR1) ... 1624
29.2.2 System Control Register 2 (SYSCR2) ... 1626
29.3 Notes on Usage ... 1628
29.3.1 Page Conflict ... 1628

Section 30 Power-Down Modes ... 1629
30.1 Features ... 1629
30.1.1 Power-Down Modes ... 1629
30.1.2 Reset ... 1630
30.2 Input/Output Pins ... 1631
30.3 Register Descriptions ... 1632
30.3.1 Standby Control Register (STBCR) ... 1632
30.3.2 Standby Control Register 2 (STBCR2) ... 1633
30.3.3 Standby Control Register 3 (STBCR3) ... 1634
30.3.4 Standby Control Register 4 (STBCR4) ... 1636
30.3.5 Standby Control Register 5 (STBCR5) ... 1637
30.3.6 Standby Control Register 6 (STBCR6) ... 1638
30.4 Operation ... 1640
30.4.1 Sleep Mode ... 1640
30.4.2 Software Standby Mode ... 1640
30.4.3 Application Example of Software Standy Mode ... 1643
30.4.4 Module Standby Function ... 1644
30.5 Usage Notes ... 1645
30.5.1 Current Consumption during Oscillation Settling Time ... 1645
30.5.2 Notes on Writing to Registers ... 1645
30.5.3 Notes on Canceling Software Standby Mode with an IRQx Interrupt Request ... 1645

Section 31 User Debugging Interface (H-UDI) 1647
31.1 Features 1647
31.2 Input/Output Pins 1649
31.3 Boundary Scan TAP Controller 1650
31.4 H-UDI TAP Controller 1653
31.5 Register Descriptions 1654
31.5.1 Bypass Register (BSBPR) 1654
31.5.2 Instruction Register (BSIR) 1654
31.5.3 ID Register (BSID) 1654
31.5.4 Boundary Scan Register (BSBSR) 1655
31.5.5 Instruction Register (SDIR) 1666
31.5.6 ID Register (SDID) 1666
31.6 Operation 1667
31.6.1 TAP Controller 1667
31.6.2 Reset Configuration 1668
31.6.3 H-UDI Reset 1668
31.6.4 H-UDI Interrupt 1669
31.6.5 Boundary Scan Operation 1669
31.7 Usage Notes 1672

Section 32 List of Registers 1675
32.1 Register Addresses (by Functional Module, in Order of the Corresponding Section Numbers) 1676
32.2 Register Bits 1704
32.3 Register States in Each Operating Mode 1744

Section 33 Electrical Characteristics 1767
33.1 Absolute Maximum Ratings 1767
33.2 DC Characteristics 1768
33.3 AC Characteristics 1772
33.3.1 Clock Timing 1773
33.3.2 Control Signal Timing 1776
33.3.3 Bus Timing 1780
33.3.4 UBC Trigger Timing 1810
33.3.5 DMAC Module Timing 1811
33.3.6 Multi Function Timer Pulse Unit 2 (MTU2) Timing 1812
33.3.7 Multi Function Timer Pulse Unit 2S (MTU2S) Timing 1814
33.3.8 POE2 Module Timing 1815
33.3.9 Watchdog Timer Timing 1816
33.3.10 Serial Communication Interface (SCI) Timing 1817
33.3.11 SCIF Module Timing 1819

33.3.12 RSPI Timing 1821
33.3.13 Controller Area Network (RCAN-ET) Timing 1825
33.3.14 IIC3 Module Timing 1826
33.3.15 A/D Trigger Input Timing 1828
33.3.16 I/O Port Timing 1829
33.3.17 EtherC Module Signal Timing 1830
33.3.18 H-UDI Related Pin Timing 1834
33.3.19 AC Characteristics Measurement Conditions 1836
33.4 A/D Converter Characteristics 1837
33.5 USB Characteristics 1838
33.6 Flash Memory Characteristics 1840
33.7 FLD Characteristics 1842
33.8 Usage Notes 1844
33.8.1 Notes on Connecting Capacitors 1844

Appendix 1845
A. Pin States 1845
B. Product Code Lineup 1855
C. Package Dimensions 1859

Main Revisions and Additions in this Edition 1863

Index 1885

Section 1 Overview

1.1 Features

This LSI is a single-chip RISC microprocessor that integrates a Renesas original RISC CPU core with peripheral functions required for system configuration.

The CPU in this LSI has a RISC-type (Reduced Instruction Set Computer) instruction set and uses a superscalar architecture and a Harvard architecture, which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture enhances data processing power. With this CPU, it has become possible to assemble low-cost, high-performance, and high-functioning systems, even for applications that were previously impossible with microprocessors, such as realtime control, which demands high speeds. This LSI also includes the floating-point unit (FPU).

In addition, this LSI includes on-chip peripheral functions necessary for system configuration, such as a large-capacity ROM, a ROM cache, a RAM, a direct memory access controller (DMAC), a data transfer controller (DTC), multi-function timer pulse units 2 (MTU2 and MTU2S), a serial communication interface with FIFO (SCIF), a serial communication interface (SCI), a Renesas serial peripheral interface (RSPI), an A/D converter, an interrupt controller (INTC), I/O ports, I^2C bus interface 3 (IIC3), a universal serial bus (USB), a controller area network (RCAN-ET), an Ethernet controller (Ether-C), and data flash (FLD).

This LSI also provides an external memory access support function to enable direct connection to various memory devices or peripheral LSIs.

These on-chip functions significantly reduce costs of designing and manufacturing application systems.

The features of this LSI are listed in table 1.1.

Table 1.1 Features

Items	Specification
CPU	• Renesas original SuperH architecture • Compatible with SH-1 and SH-2 at object code level • 32-bit internal data bus • Support of an abundant register-set Sixteen 32-bit general registers Four 32-bit control registers Four 32-bit system registers Register bank for high-speed response to interrupts • RISC-type instruction set (upward compatible with SH series) Instruction length: 16-bit fixed-length basic instructions for improved code efficiency and 32-bit instructions for high performance and usability Load/store architecture Delayed branch instructions Instruction set based on C language • Superscalar architecture to execute two instructions at one time • Instruction execution time: Up to two instructions/cycle • Address space: 4 Gbytes • Internal multiplier • Five-stage pipeline • Harvard architecture

Items	Specification
FPU (SH7216 Group only)	• On-chip floating-point coprocessor • Supports single-precision (32 bits) and double-precision (64 bits) • Supports IEEE 754-compliant data types and exceptions • Rounding mode: Round to Nearest and Round to Zero • Handling of denormalize numbers: Truncation to Zero • Floating-point registers Sixteen 32-bit floating-point registers (single-precision x 16 words or double-precision x 8 words) Two 32-bit floating-point system registers • Supports FMAC (multiply and accumulate) instruction • Supports FDIV (division) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution times Latency (FMAC/FADD/FSUB/FMUL): 3 cycles (single-precision), 8 cycles (double-precision) Pitch (FMAC/FADD/FSUB/FMUL): 1 cycle (single-precision), 6 cycles (double-precision) Note: FMAC is supported for single-precision only. • Five-stage pipeline
Operating modes	• Operating modes Extended ROM enabled mode Single-chip mode • Processing states Program execution state Exception handling state Bus mastership release state • Power-down modes Sleep mode Software standby mode Module standby mode

Items	Specification
ROM cache	• Instruction/data separation system • Instruction prefetch cache: Full/set associative • Instruction prefetch miss cache: Full/set associative • Data cache: Full/set associative • Line size: 16 bytes • Hardware prefetch function (continuous/branch prefetch)
Interrupt controller (INTC)	• Nine external interrupt pins (NMI and IRQ7 to IRQ0) • On-chip peripheral interrupts: Priority level set for each module • 16 priority levels available • Register bank enabling fast register saving and restoring in interrupt processing
Bus state controller (BSC)	• Address space divided into eight areas (0 to 7), each a maximum of 64 Mbytes a Harvard architecture • External bus: 8, 16, or 32 bits • The following features settable for each area independently Supports both big endian and little endian for data access Bus size (8, 16, or 32 bits): Available sizes depend on the area. Number of access wait cycles (different wait cycles can be specified for read and write access cycles in some areas) Idle wait cycle insertion (between same area access cycles or different area access cycles) • SDRAM refresh Auto refresh or self refresh mode selectable • SDRAM burst access
Direct memory access controller (DMAC)	• Eight channels; external request available for four channels of them • Can be activated by on-chip peripheral modules • Burst mode and cycle steal mode • Intermittent mode available (16 and 64 cycles supported) • Transfer information can be automatically reloaded

Items	Specification
Data transfer controller (DTC)	• Data transfer activated by an on-chip peripheral module interrupt can be done independently of the CPU transfer. • Transfer mode selectable for each interrupt source (transfer mode is specified in memory) • Multiple data transfer enabled for one activation source • Various transfer modes Normal mode, repeat mode, or block transfer mode can be selected. • Data transfer size can be specified as byte, word, or longword • The interrupt that activated the DTC can be issued to the CPU. A CPU interrupt can be requested after one data transfer completion. • A CPU interrupt can be requested after all specified data transfer completion.
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EXTAL) or crystal resonator • Input clock can be multiplied by 16 by the internal PLL circuit • Five types of clocks generated: CPU clock: Maximum 200 MHz (SH7216A, SH7216B, SH7214A, and SH7214B) Maximum 100 MHz (SH7216G, SH7216H, SH7214G, and SH7214H) Bus clock: Maximum 50 MHz Peripheral clock: Maximum 50 MHz Timer clock: Maximum 100 MHz AD clock: Maximum 50 MHz
Watchdog timer (WDT)	• On-chip one-channel watchdog timer • A counter overflow can reset the LSI
Power-down modes	• Three power-down modes provided to reduce the current consumption in this LSI Sleep mode Software standby mode Module standby mode

Items	Specification
Multi-function timer pulse unit 2 (MTU2)	• Maximum 16 lines of pulse input/output and 3 lines of pulse input based on six channels of 16-bit timers • 21 output compare and input capture registers • Input capture function • Pulse output modes Toggle, PWM, and complementary PWM • Synchronization of multiple counters • Complementary PWM output mode Non-overlapping waveforms output for 3-phase inverter control Automatic dead time setting 0% to 100% PWM duty value specifiable A/D conversion delaying function Interrupt skipping at crest or trough • Reset-synchronized PWM mode Three-phase PWM waveforms in positive and negative phases can be output with a required duty value • Phase counting mode Two-phase encoder pulse counting available
Multi-function timer pulse unit 2S (MTU2S)	• Subset of MTU2, included in channels 3 to 5 • Operating at 100 MHz max.
Port output enable 2 (POE2)	• High-impedance control of high-current pins at a falling edge or low-level input on the $\overline{\text{POE}}$ pin
Compare match timer (CMT)	• Two-channel 16-bit counters • Four types of clock can be selected (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) • DMA transfer request or interrupt request can be issued when a compare match occurs
Serial communication interface (SCI)	• Four channels • Clocked synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator

Items	Specification
Serial communication interface with FIFO (SCIF)	• One channel • Clocked synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • Separate 16-byte FIFO registers for transmission and reception
Renesas serial peripheral interface (RSPI)	• Clock synchronous mode serial communications • Master mode or slave mode selectable • Modifiable bit length, clock polarity, and clock phase • A transfer can be executed in sequential loops • Switchable MSB first/LSB first • Maximum transfer rate: 12.5 MHz • Up to four slaves can be controlled in single master mode (depends on the PFC setting) • Up to three slaves can be controlled in multi-master mode (depends on the PFC setting)
Universal serial bus (USB)	• USB 2.0 full-speed mode (12 Mbps) supported • On-chip bus transceiver • Standard commands automatically processed by hardware • Three transfer modes (control transfer, balk transfer, and interrupt transfer) • 27 types of interrupt sources available • DMA transfer interface • EP1 to EP9: assigned to Bulk IN, Bulk OUT, or Interrupt IN

Items	Specification
Ethernet controller (EtherC) (SH7216A, SH7214A, SH7216G, SH7214G)	• Media Access Control function (MAC) Assembling or disassembling data frames (the format based on IEEE 802.3) Link management using CSMA/CD (to prevent collision and process when a collision occurs) CRC processing FIFO (2 Kbytes each for transmission and reception) Full-duplex and half-duplex sending/receiving available Conforms to IEEE802.3x flow control (back pressure) • Supports the MII (Media Independent Interface) standard Station management (STA function) Transfer rate: 10/100 Mbps • Magic Packet (supports Wake On LAN (WOL) output)
DMAC for Ethernet controller (E-DMAC) (SH7216A, SH7214A, SH7216G, SH7214G)	• CPU load reduced by descriptor management • One transfer channel from EtherC receive FIFO to the receive buffer • One transfer channel from the send buffer to EtherC transmit FIFO • System bus efficiently used by 32-byte burst transfer • Supports single-frame and multi-buffer operation
Controller area network (RCAN-ET)	• CAN version: Bosch 2.0B active is supported • Buffer size: 15 buffers for transmission/reception and one buffer for reception only • One channel
I^2C bus interface 3 (IIC3)	• One channel • Master mode and slave mode supported
I/O ports	• Input or output can be selected for each bit
A/D converter	• Two modules • 12-bit resolution • Eight input channels • Sampling can be carried out simultaneously on three channels. • A/D conversion request by the external trigger or timer trigger
ASE break controller (ABC)	• Ten break channels • The cycle of the internal bus can be set as break conditions

Items	Specification
User break controller (UBC)	• Four break channels • Addresses, data values, type of access, and data size can all be set as break conditions
User debugging interface (H-UDI)	• E10A emulator support • JTAG-standard pin assignment • Boundary scan test port conforming to IEEE 1149.1 • Realtime branch trace
Advanced user debugger (AUD)	• Six input/output pins • Branch source address/destination address trace • Window data trace • Full trace All trace data can be output by interrupting CPU operation • Realtime trace Trace data can be output within the range where CPU operation is not interrupted
On-chip ROM	• 1 Mbyte, 768 Kbytes, 512 Kbytes
On-chip RAM	• Eight pages, six pages, four pages • 128 Kbytes, 96 Kbytes, 64 Kbytes
Data flash (FLD)	• 32 Kbytes • Programmed in 8-byte units
Power supply voltage	• $V_{CC}Q$: 3.0 to 3.6 V, AV_{CC}: 4.5 to 5.5 V
Packages	• PLQP0176KB-A (0.5-mm pitch) • PLQP0176LB-A (0.4-mm pitch) • PLBG0176GA-A (0.8-mm pitch)

1.2 Block Diagram

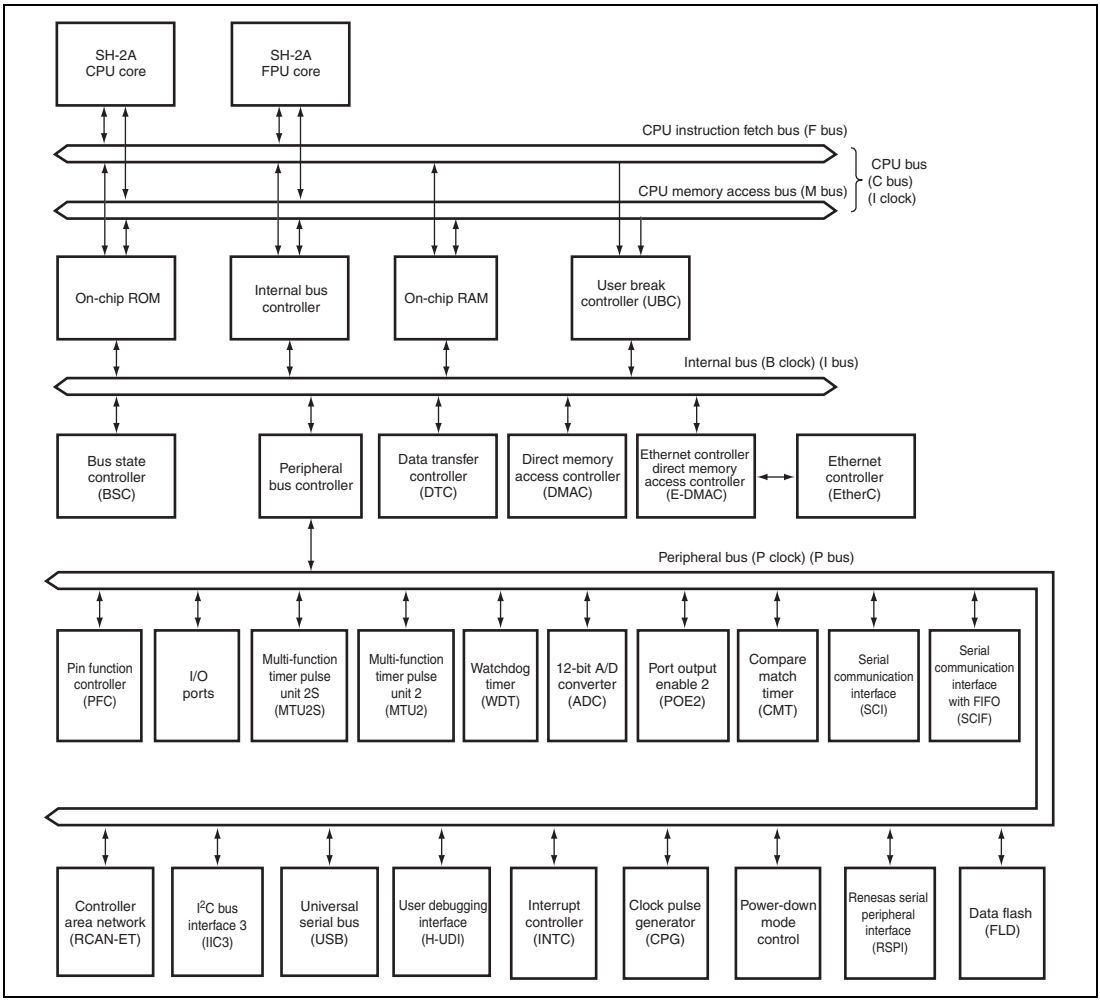


Figure 1.1 Block Diagram

1.3 Pin Assignment

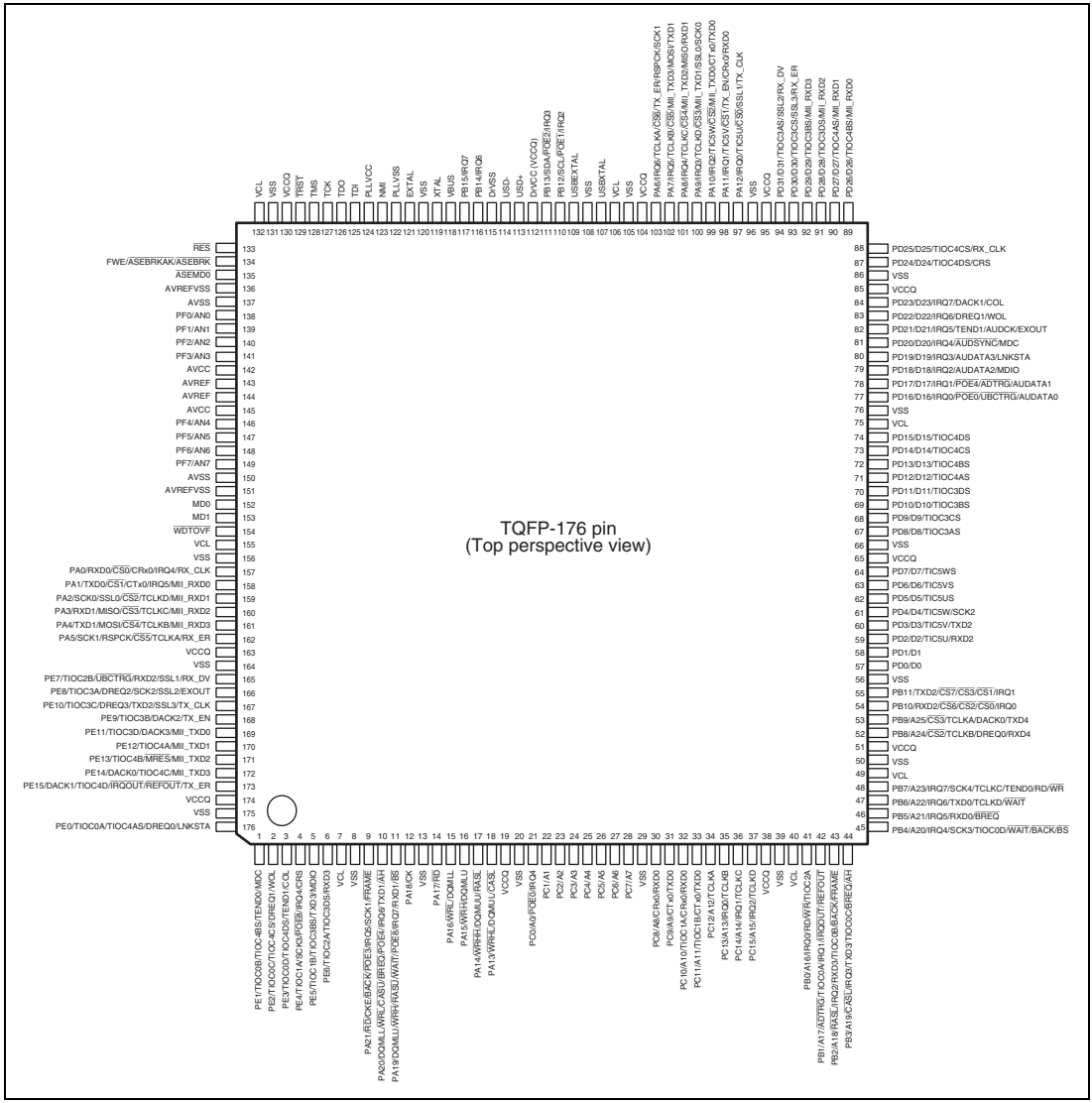


Figure 1.2 Pin Assignment (1)

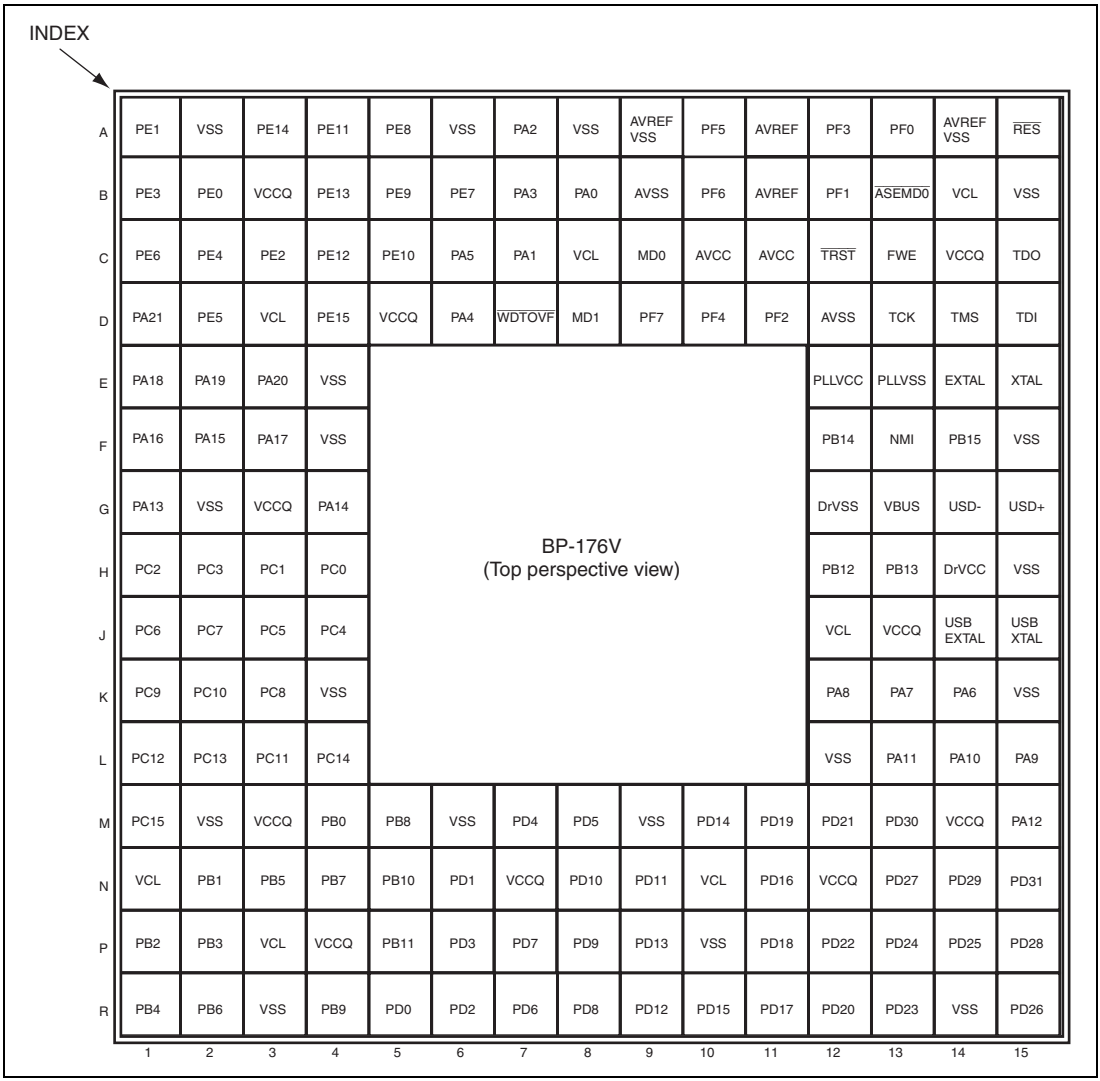


Figure 1.3 Pin Assignment (2)

1.4 Pin Functions

Table 1.2 lists functions of each pin.

Table 1.2 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	V_{CL}	Input	Internal step-down power supply	External capacitance pins for internal step-down power supply. All the V_{CL} pins must be connected to the V_{SS} pins via a 0.1-μF capacitor (should be placed close to the pins). The system power supply must not be directly connected to the V_{CL} pins.
	V_{SS}	Input	Ground	Ground pins. All the V_{SS} pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	$V_{CC}Q$	Input	Power supply	Power supply pins. All the $V_{CC}Q$ pins must be connected to the system power supply. This LSI does not operate if there is a pin left open.
	$PLLV_{CC}$	Input	PLL power supply	Power supply for the on-chip PLL oscillator. Apply the same electric potential as that on the $V_{CC}Q$ pin.
	$PLLV_{SS}$	Input	Ground for PLL	Ground pin for the on-chip PLL oscillator.
Clock	EXTAL	Input	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	Output	Crystal	Connected to a crystal resonator.
	USBEXTAL	Input	Crystal for USB	Connected to a resonator for the USB.
	USBXTAL	Output	Crystal for USB	Connected to a resonator for the USB.
	CK	Output	System clock	Supplies the system clock to external devices.

Classification	Symbol	I/O	Name	Function
Operating mode control	MD1, MD0	Input	Mode set	Sets the operating mode. Do not change the signal levels on these pins during operation.
	$\overline{\text{ASEMD0}}$	Input	Debugging mode	Enables the E10A-USB emulator functions. Input a high level to operate the LSI in normal mode (not in debugging mode). To operate it in debugging mode, apply a low level to this pin on the user system board.
	FWE	Input	Flash memory write enable	Pin for flash memory. Flash memory can be protected against writing or erasure through this pin.
System control	$\overline{\text{RES}}$	Input	Power-on reset	This LSI enters the power-on reset state when this signal goes low.
	$\overline{\text{MRES}}$	Input	Manual reset	This LSI enters the manual reset state when this signal goes low.
	$\overline{\text{WDTOVF}}$	Output	Watchdog timer overflow	Outputs an overflow signal from the WDT. Use a resistor with a value of at least 1 MΩ to pull this pin down.
	$\overline{\text{BREQ}}$	Input	Bus-mastership request	A low level is input to this pin when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	Output	Bus-mastership request acknowledge	Indicates that the bus mastership has been released to an external device. Reception of the $\overline{\text{BACK}}$ signal informs the device which has output the $\overline{\text{BREQ}}$ signal that it has acquired the bus.

Classification	Symbol	I/O	Name	Function
Interrupts	NMI	Input	Non-maskable interrupt	Non-maskable interrupt request pin. Fix it high when not in use.
	IRQ7 to IRQ0	Input	Interrupt requests 7 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising edge, falling edge, or both edges can also be selected.
	$\overline{\text{IRQOUT}}$	Output	Interrupt request output	Indicates that an interrupt has occurred, enabling external devices to be informed of an interrupt occurrence even while the bus mastership is released.
Address bus	A25 to A0	Output	Address bus	Outputs addresses.
Data bus	D31 to D0	I/O	Data bus	Bidirectional data bus.
Bus control	$\overline{\text{CS7}}$ to $\overline{\text{CS0}}$	Output	Chip select 7 to 0	Chip-select signals for external memory or devices.
	$\overline{\text{RD}}$	Output	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	Output	Read/write	Read/write signal.
	$\overline{\text{BS}}$	Output	Bus start	Bus-cycle start signal.
	$\overline{\text{AH}}$	Output	Address hold	Address hold timing signal for the device that uses the address/data-multiplexed bus.
	$\overline{\text{WAIT}}$	Input	Wait	Input signal for inserting a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{FRAME}}$	Output	Frame signal	In burst MPX-I/O interface mode, negated before the last bus cycle to indicate that the next bus cycle is the last access
	$\overline{\text{WRHH}}$	Output	Write to HH byte	Indicates a write access to bits 31 to 24 of data of external memory or device.
	$\overline{\text{WRHL}}$	Output	Write to HL byte	Indicates a write access to bits 23 to 16 of data of external memory or device.

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{\text{WRH}}$	Output	Write to upper byte	Indicates a write access to bits 15 to 8 of data of external memory or device.
	$\overline{\text{WRL}}$	Output	Write to lower byte	Indicates a write access to bits 7 to 0 of data of external memory or device.
	DQMUU	Output	HH byte selection	Selects bits D31 to D24 when SDRAM is connected.
	DQMUL	Output	HL byte selection	Selects bits D23 to D16 when SDRAM is connected.
	DQMLU	Output	Upper byte selection	Selects bits D15 to D8 when SDRAM is connected.
	DQMLL	Output	Lower byte selection	Selects bits D7 to D0 when SDRAM is connected.
	$\overline{\text{RASU}}$	Output	RAS	Connected to the $\overline{\text{RAS}}$ pin when SDRAM is connected.
	$\overline{\text{CASU}}$	Output	CAS	Connected to the $\overline{\text{CAS}}$ pin when SDRAM is connected.
	$\overline{\text{RASL}}$	Output	RAS	Connected to the $\overline{\text{RAS}}$ pin when SDRAM is connected.
	$\overline{\text{CASL}}$	Output	CAS	Connected to the $\overline{\text{CAS}}$ pin when SDRAM is connected.
	CKE	Output	CK enable	Connected to the CKE pin when SDRAM is connected.
	$\overline{\text{REFOUT}}$	Output	Refresh request output	Request signal output for refresh execution while the bus mastership is released.
Direct memory access controller (DMAC)	DREQ0 to DREQ3	Input	DMA-transfer request	Input pins to receive external requests for DMA transfer.
	DACK0 to DACK3	Output	DMA-transfer request accept	Output pins for signals indicating acceptance of external requests from external devices.
	TEND1, TEND0	Output	DMA-transfer end output	Output pins for DMA transfer end.

Classification	Symbol	I/O	Name	Function
Multi-function timer pulse unit 2 (MTU2)	TCLKA, TCLKB, TCLKC, TCLKD	Input	MTU2 timer clock input	External clock input pins for the timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	MTU2 input capture/output compare (channel 0)	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A, TIOC1B	I/O	MTU2 input capture/output compare (channel 1)	The TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A, TIOC2B	I/O	MTU2 input capture/output compare (channel 2)	The TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	MTU2 input capture/output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	MTU2 input capture/output compare (channel 4)	The TGRA_4 and TGRD_4 input capture input/output compare output/PWM output pins.
	TIC5U, TIC5V, TIC5W	Input	MTU2 input capture (channel 5)	The TGRU_5, TGRV_5, and TGRW_5 input capture input/dead time compensation input pins.
Port output enable 2 (POE2)	$\overline{\text{POE8}}$, $\overline{\text{POE4}}$ to $\overline{\text{POE0}}$	Input	Port output control	Request signal input to place the MTU2 and MTU2S waveform output pin in the high impedance state.

Classification	Symbol	I/O	Name	Function
Multi-function timer pulse unit 2S (MTU2S)	TIOC3AS, TIOC3BS, TIOC3CS, TIOC3DS	I/O	MTU2S input capture/output compare (channel 3)	The TGRA_3S to TGRD_3S input capture input/output compare output/PWM output pins.
	TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS	I/O	MTU2S input capture/output compare (channel 4)	The TGRA_4S and TGRD_4S input capture input/output compare output/PWM output pins.
	TIOC5US, TIOC5VS, TIOC5WS	Input	MTU2S input capture (channel 5)	The TGRU_5S, TGRV_5S, and TGRW_5S input capture input/dead time compensation input pins.
Serial communication interface (SCI)	TXD4, TXD2 to TXD0	Output	Transmit data	Data output pins.
	RXD4, RXD2 to RXD0	Input	Receive data	Data input pins.
	SCK4, SCK2 to SCK0	I/O	Serial clock	Clock input/output pins.
Serial communication interface with FIFO (SCIF)	TXD3	Output	Transmit data	Data output pin.
	RXD3	Input	Receive data	Data input pin.
	SCK3	I/O	Serial clock	Clock input/output pin.
Renesas serial peripheral interface (RSPI)	MOSI	I/O	Data	Data input/output pin.
	MISO	I/O	Data	Data input/output pin.
	RSPCK	I/O	Clock	Clock input/output pin.
	SSL0	I/O	Chip select	Chip select input/output pin.
	SSL1 to SSL3	Output		

Classification	Symbol	I/O	Name	Function
Universal serial bus (USB)	DrV_{CC} ($V_{CC}Q$)	Input	USB power supply	Power supply pin for the internal transceiver. Connect it to the system power supply. Must be at same potential as $V_{CC}Q$.
	DrV_{SS}	Input	USB ground	Ground pin for the internal transceiver.
	USD+, USD−	I/O	USB data	USB data input/output pins.
	VBUS	Input	Cable connection monitor	USB cable connection monitor input pin.
	PUPD (PB15)	Output	Pull-up control	Use this pin for the pull-up control of USD+ signal
Controller area network (RCAN-ET)	CTx0	Output	Transmit data	Transmit data pin for CAN bus.
	CRx0	Input	Receive data	Receive data pin for CAN bus.
I^2C bus interface 3 (IIC3)	SCL	I/O	Serial clock pin	Serial clock input/output pin.
	SDA	I/O	Serial data pin	Serial data input/output pin.
A/D converter	AN7 to AN0	Input	Analog input pins	Analog input pins.
	$\overline{ADTRG}$	Input	A/D conversion trigger input	External trigger input pin for starting A/D conversion.
	AV_{CC}	Input	Analog power supply	Power supply pin for the A/D converter. Connect this pin to the system power supply ($V_{CC}Q$) when the A/D converter is not used.
	AV_{REF}	Input	Analog reference power supply	Reference voltage pin for the A/D converter.
	AV_{SS}	Input	Analog ground	Ground pin for the A/D converter. Connect this pin to the system power supply (V_{SS}) when the A/D converter is not used.
	$AV_{REF}V_{SS}$	Input	Analog reference ground	Reference ground pin for the A/D converter. Connect this pin to the system power supply (V_{SS}) when the A/D converter is not used.

Classification	Symbol	I/O	Name	Function
Ethernet controller (EtherC)	CRS	Input	Carrier sense	Carrier sense signal input.
	COL	Input	Collision	Signal collision detection signal input.
	MII_TXD3 to MII_TXD0	Output	Transmit data	4-bit transmit data.
	TX_EN	Output	Transmit enable	Indicates that transmit data is ready on MII_TXD3 to MII_TXD0 pins.
	TX_CLK	Input	Transmit clock	Timing input as reference for TX_EN, TX_ER, and MII_TXD3 to MII_TXD0.
	TX_ER	Output	Transmit error	Notifies PHY_LSI of error during transmission.
	MII_RXD3 to MII_RXD0	Input	Receive data	4-bit receive data.
	RX_DV	Input	Receive data valid	Indicates that there is valid receive data on MII_RXD3 to MII_RXD0 pins.
	RX_CLK	Input	Receive clock	Timing input as reference for RX_DV, RX_ER, and MII_RXD3 to MII_RXD0.
	RX_ER	Input	Receive error	Recognizes the error during reception.
	MDC	Output	Management clock	Clock signal for information transfer via MDIO.
	MDIO	I/O	Management data	Bidirectional data for exchange of management information.
	WOL	Output	MAGIC packet receive	Receives Magic packets.
	LNKSTA	Input	Link status	Inputs link status from the PHY-LSI.
	EXOUT	Output	General output	External output.

Classification	Symbol	I/O	Name	Function
I/O ports	PA21 to PA0	I/O	General port	22-bit general input/output port pins.
	PB15 to PB0	I/O	General port	14-bit general input/output port and 2-bit general input port pins.
	PC15 to PC0	I/O	General port	16-bit general input/output port pins.
	PD31 to PD0	I/O	General port	32-bit general input/output port pins.
	PE15 to PE0	I/O	General port	16-bit general input/output port pins.
	PF7 to PF0	Input	General port	8-bit general input port pins.
User debugging interface (H-UDI)	TCK	Input	Test clock	Test-clock input pin.
	TMS	Input	Test mode select	Test-mode select signal input pin.
	TDI	Input	Test data input	Serial input pin for instructions and data.
	TDO	Output	Test data output	Serial output pin for instructions and data.
	$\overline{\text{TRST}}$	Input	Test reset	Initialization-signal input pin. Input a low level when not using the H-UDI.
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	Output	AUD data	Branch destination/source address output pin
	AUDCK	Output	AUD clock	Sync clock output pin
	$\overline{\text{AUDSYNC}}$	Output	AUD sync signal	Data start-position acknowledge-signal output pin
Emulator interface	$\overline{\text{ASEBRKAK}}$	Output	Break mode acknowledge	Indicates that the E10A-USB emulator has entered its break mode.
	$\overline{\text{ASEBRK}}$	Input	Break request	E10A-USB emulator break input pin.
User break controller (UBC)	$\overline{\text{UBCTRG}}$	Output	User break trigger output	Trigger output pin for UBC condition match.

Section 2 CPU

2.1 Data Formats

Figure 2.1 shows the data formats supported by the SH-2A/SH2A-FPU.

The CPU of SH7216 Group products (SH7216A, SH7216B, SH7216G and SH7216H) is the SH2A-FPU, and that of SH7214 Group products (SH7214A, SH7214B, SH7214G and SH7214H) is the SH-2A.

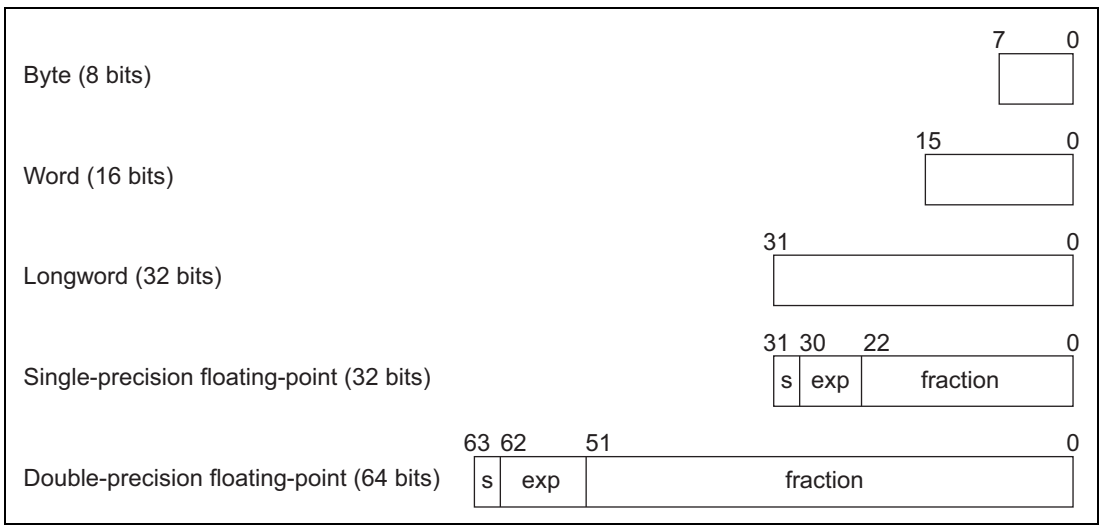


Figure 2.1 Data Format

2.2 Register Descriptions

2.2.1 General Registers

Figure 2.2 shows the general registers.

The general registers consist of 16 registers, numbered R0 to R15, and are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and restoring the status register (SR) and program counter (PC) in exception handling is accomplished by referencing the stack using R15.

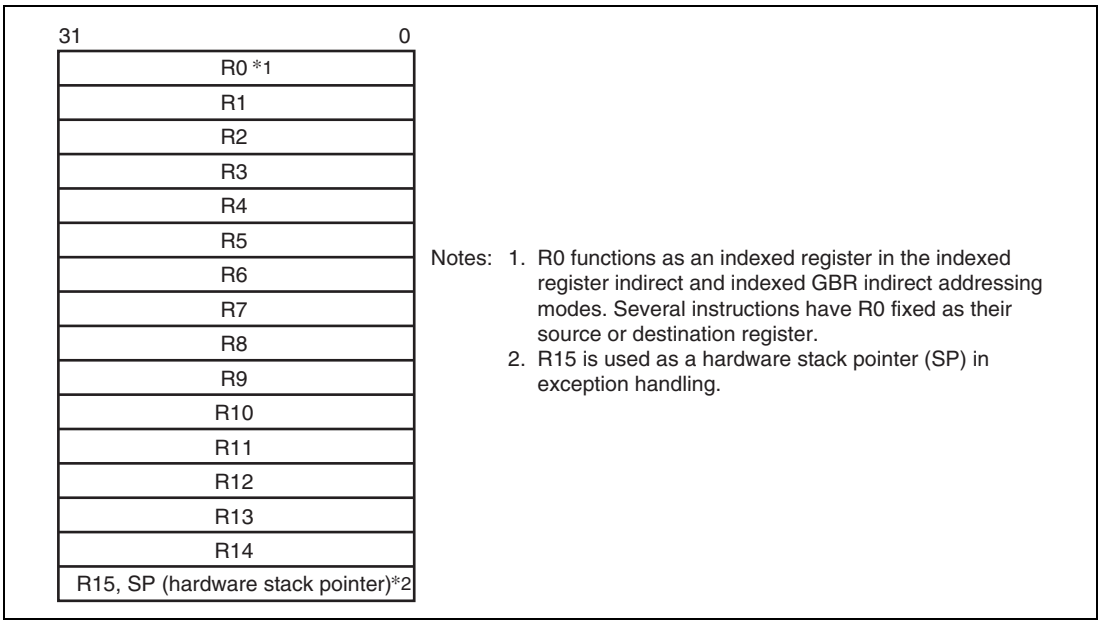


Figure 2.2 General Registers

2.2.2 Control Registers

The control registers consist of four 32-bit registers: the status register (SR), the global base register (GBR), the vector base register (VBR), and the jump table base register (TBR).

The status register indicates instruction processing states.

The global base register functions as a base address for the GBR indirect addressing mode to transfer data to the registers of on-chip peripheral modules.

The vector base register functions as the base address of the exception handling vector area (including interrupts).

The jump table base register functions as the base address of the function table area.

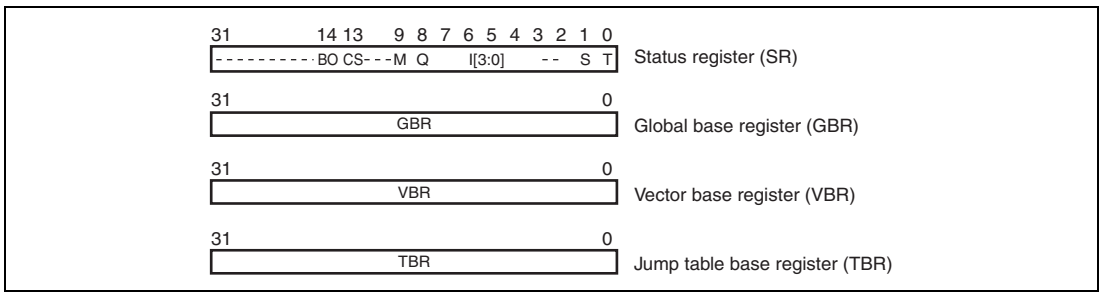


Figure 2.3 Control Registers

(1) Status Register (SR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-	S	T
Initial value:	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W:	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14	BO	0	R/W	BO Bit Indicates the register bank has overflowed.
13	CS	0	R/W	CS Bit Indicates, in CLIP instruction execution, the value has exceeded the saturation upper-limit value or fallen below the saturation lower-limit value.
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	M	—	R/W	M Bit
8	Q	—	R/W	Q Bit Used by the DIV0S, DIV0U, and DIV1 instructions.
7 to 4	I[3:0]	1111	R/W	Interrupt Mask Level
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	S	—	R/W	S Bit Specifies a saturation operation for a MAC instruction.
0	T	—	R/W	T Bit True/false condition or carry/borrow bit

(2) Global Base Register (GBR)

GBR is referenced as the base address in a GBR-referencing MOV instruction.

(3) Vector Base Register (VBR)

VBR is referenced as the branch destination base address when an exception or an interrupt occurs.

(4) Jump Table Base Register (TBR)

TBR is referenced as the start address of a function table located in memory in a JSR/N@@(disp8,TBR) table-referencing subroutine call instruction.

2.2.3 System Registers

The system registers consist of four 32-bit registers: the high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). MACH and MACL store the results of multiply or multiply and accumulate operations. PR stores the return address from a subroutine procedure. PC indicates the program address being executed and controls the flow of the processing.

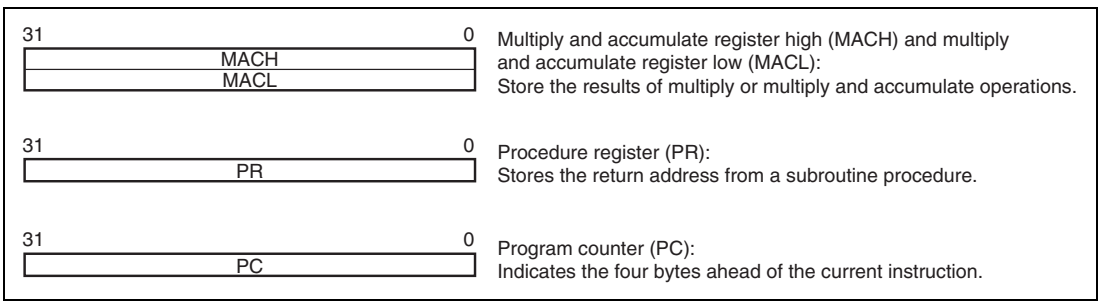


Figure 2.4 System Registers

(1) Multiply and Accumulate Register High (MACH) and Multiply and Accumulate Register Low (MACL)

MACH and MACL are used as the addition value in a MAC instruction, and store the result of a MAC or MUL instruction.

(2) Procedure Register (PR)

PR stores the return address of a subroutine call using a BSR, BSRF, or JSR instruction, and is referenced by a subroutine return instruction (RTS).

(3) Program Counter (PC)

PC indicates the address four bytes farther from that of the instruction being executed.

2.2.4 Floating-Point Registers

Figure 2.5 shows the floating-point registers. There are sixteen 32-bit floating-point registers, FPR0 to FPR15. These sixteen registers are referenced as FR0 to FR15, DR0, DR2, DR4, DR6, DR8, DR10, DR12, and DR14. The correspondence between FPRn and the referenced name is determined by the PR and SZ bits in FPSCR (see figure 2.5).

(1) Floating-Point Registers (FPRn: 16 registers)

FPR0, FPR1, FPR2, FPR3, FPR4, FPR5, FPR6, FPR7, FPR8, FPR9, FPR10, FPR11, FPR12, FPR13, FPR14, and FPR15

(2) Single-Precision Floating-Point Registers (FRi: 16 registers)

FR0 to FR15 are allocated to FPR0 to FPR15.

(3) Double-Precision Floating-Point Registers or Single-Precision Floating-Point Register Pairs (DRi: 8 registers)

A DR register is composed of two FR registers.

DR0 = {FPR0, FPR1}, DR2 = {FPR2, FPR3}, DR4 = {FPR4, FPR5}, DR6 = {FPR4, FPR5}, DR8 = {FPR8, FPR9}, DR10 = {FPR10, FPR11}, DR12 = {FPR12, FPR13}, and DR14 = {FPR14, FPR15}

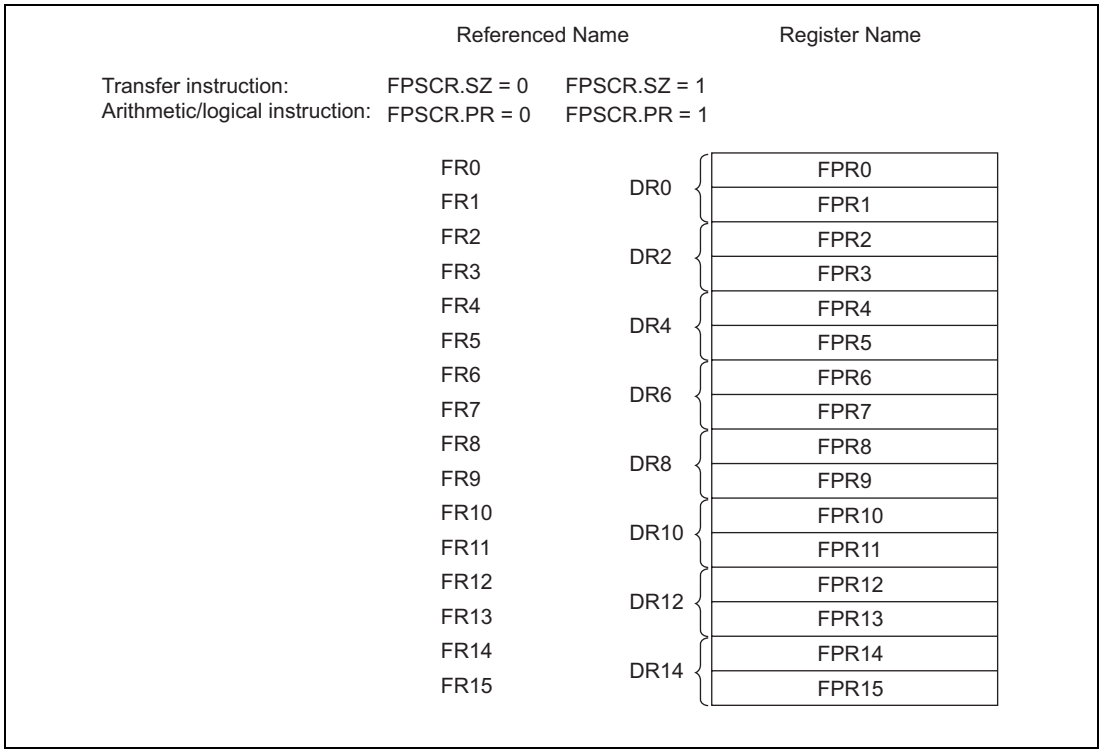


Figure 2.5 Floating-Point Registers

Programming Note: The values of FPR0 to FPR15 are undefined after a reset.

2.2.5 Floating-Point System Registers

(1) Floating-Point Communication Register (FPUL)

Data is transferred between an FPU register and a CPU register via FPUL.

(2) Floating Point Status/Control Register (FPSCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	QIS	-	SZ	PR	DN	Cause	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable					Flag					RM[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	QIS	0	R/W	sNaN is treated as qNaN or ±∞. Valid only when the V bit in the FPU exception enable field (Enable) is set to 1. 0: Processed as qNaN or ±∞ 1: Exception generated (processed same as sNaN)
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	SZ	0	R/W	Transfer Size Mode 0: Sets the size of an FMOV instruction to 32 bits. 1: Sets the size of an FMOV instruction to 32-bit pair (64 bits).
19	PR	0	R/W	Precision Mode 0: Executes floating-point instructions in single precision. 1: Executes floating-point instructions in double precision (the result of an instruction with no support for double-precision is undefined).

Bit	Bit Name	Initial Value	R/W	Description
18	DN	1	R/W	Denormalization Mode This bit is always set to 1. 1: A denormalized number is treated as zero.
17 to 12	Cause	All 0	R/W	FPU exception cause field
11 to 7	Enable	All 0	R/W	FPU exception enable field
6 to 2	Flag	All 0	R/W	FPU exception flag field When an FPU operation instruction is first executed, the FPU exception cause field is set to 0; when an FPU exception next occurs, the corresponding bit in the FPU exception cause field and FPU exception flag field is set to 1. The FPU exception flag field retains the status of an exception generated after that field was last cleared. For bit allocation for each field, see table 2.1.
1, 0	RM[1:0]	01	R/W	Round Mode 00: Round to nearest 01: Round to zero 10: Reserved 11: Reserved

Table 2.1 Bit Allocation for FPU Exception Handling

		FPU Error (E)	Invalid Operation (V)	Division by 0 (Z)	Overflow (O)	Underflow (U)	Incorrect (I)
Cause	FPU exception cause field	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12
Enable	FPU exception enable field	None	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7
Flag	FPU exception flag field	None	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

Note: In the SH-2A, no FPU errors occur.

2.2.6 Register Bank

Using a register bank, high-speed register saving and restoration can be achieved for the 19 32-bit registers: general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses the bank. Restoration from the bank is executed by a RESBANK instruction issued in an interrupt processing routine.

This LSI has 15 banks. For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.2.7 Initial Values of Registers

Table 2.2 lists the values of the registers after a reset.

Table 2.2 Initial Values of Registers

Classification	Register	Initial Value
General registers	R0 to R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I[3:0] are 1111 (H'F), BO and CS are 0, reserved bits are 0, and others are undefined
	GBR, TBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table
Floating-point registers	FPR0 to FPR15	Undefined
Floating-point system registers	FPUL	Undefined
	FPSCR	H'00040001

2.3 Data Formats

2.3.1 Data Format in Registers

Register operands are always longwords (32 bits). If the size of a memory operand is a byte (8 bits) or a word (16 bits), it is changed into a longword through sign extension or zero extension when loaded into a register.

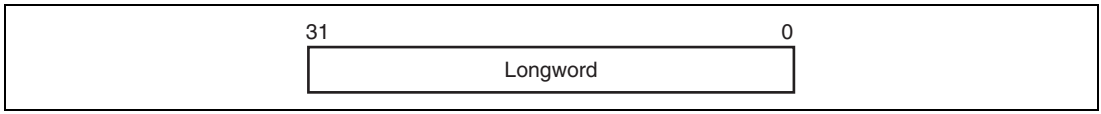


Figure 2.6 Data Format in Registers

2.3.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Memory can be accessed in 8-bit bytes, 16-bit words, or 32-bit longwords. A memory operand of fewer than 32 bits is stored in a register in sign-extended or zero-extended form.

A word operand should be accessed at a word boundary (an even address of multiple of two bytes: address 2n), and a longword operand at a longword boundary (an even address of multiple of four bytes: address 4n). Otherwise, an address error will occur. A byte operand can be accessed at any address.

Only big-endian byte order can be selected for the data format.

Data formats in memory are shown in figure 2.7.

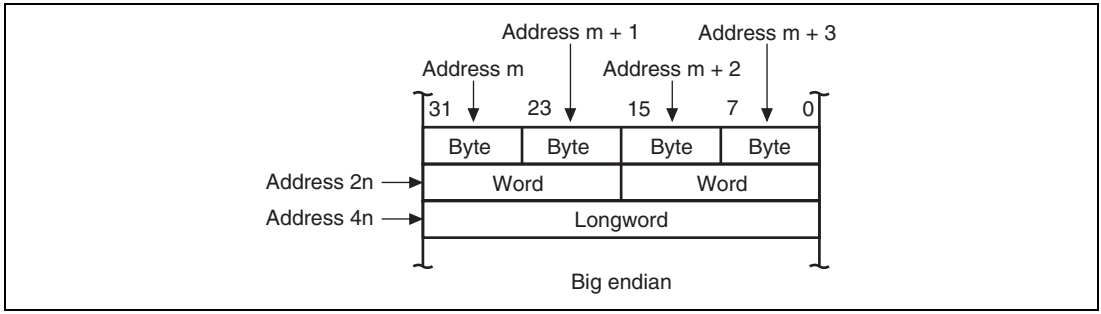


Figure 2.7 Data Formats in Memory

2.3.3 Immediate Data Format

Byte (8-bit) immediate data is located in an instruction code. Immediate data accessed by the MOV, ADD, and CMP/EQ instructions is sign-extended and handled in registers as longword data. Immediate data accessed by the TST, AND, OR, and XOR instructions is zero-extended and handled as longword data. Consequently, AND instructions with immediate data always clear the upper 24 bits of the destination register.

20-bit immediate data is located in the code of a MOVI20 or MOVI20S 32-bit transfer instruction. The MOVI20 instruction stores immediate data in the destination register in sign-extended form. The MOVI20S instruction shifts immediate data by eight bits in the upper direction, and stores it in the destination register in sign-extended form.

Word or longword immediate data is not located in the instruction code, but rather is stored in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

See examples given in section 2.4.1 (10), Immediate Data.

2.4 Instruction Features

2.4.1 RISC-Type Instruction Set

The CPU has a RISC-type instruction set, which features following functions.

(1) 16-Bit Fixed-Length Instructions

Basic instructions have a fixed length of 16 bits, improving program code efficiency.

(2) 32-Bit Fixed-Length Instructions

The SH-2A/SH2A-FPU additionally features 32-bit fixed-length instructions, improving performance and ease of use.

(3) One Instruction per Cycle

Each basic instruction can be executed in one cycle using the pipeline system.

(4) Data Length

The standard data length for all operations is a longword. Memory can be accessed in bytes, words, or longwords. Byte or word data in memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It is also handled as longword data.

Table 2.3 Sign Extension of Word Data

SH2-A/SH2A-FPU CPU	Description	Example of Other CPU

`MOV.W @(disp,PC),R1`
`ADD R1,R0`
`.........`
`.DATA.W H'1234`
	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	`ADD.W #H'1234,R0`

Note: @(disp, PC) accesses the immediate data.

(5) Load-Store Architecture

Basic operations are executed between registers. For operations that involve memory access, data is loaded to the registers and executed (load-store architecture). Instructions such as AND that manipulate bits, however, are executed directly in memory.

(6) Delayed Branch Instructions

With the exception of some instructions, unconditional branch instructions, etc., are executed as delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction immediately following the delayed branch instruction. This reduces disturbance of the pipeline control when a branch is taken.

In a delayed branch, the actual branch operation occurs after execution of the slot instruction. However, instruction execution such as register updating excluding the actual branch operation, is performed in the order of delayed branch instruction → delay slot instruction. For example, even though the contents of the register holding the branch destination address are changed in the delay slot, the branch destination address remains as the register contents prior to the change.

Table 2.4 Delayed Branch Instructions

SH2-A/SH2A-FPU CPU		Description	Example of Other CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

(7) Unconditional Branch Instructions with No Delay Slot

The SH-2A/SH2A-FPU additionally features unconditional branch instructions in which a delay slot instruction is not executed. This eliminates unnecessary NOP instructions, and so reduces the code size.

(8) Multiply/Multiply-and-Accumulate Operations

16-bit × 16-bit → 32-bit multiply operations are executed in one to two cycles. 16-bit × 16-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to three cycles. 32-bit × 32-bit → 64-bit multiply and 32-bit × 32-bit + 64-bit → 64-bit multiply-and-accumulate operations are executed in two to four cycles.

(9) T Bit

The T bit in the status register (SR) changes according to the result of the comparison. Whether a conditional branch is taken or not taken depends upon the T bit condition (true/false). The number of instructions that change the T bit is kept to a minimum to improve the processing speed.

Table 2.5 T Bit

SH2-A/SH2A-FPU CPU		Description	Example of Other CPU	
CMP/GE	R1,R0	T bit is set when R0 ≥ R1.	CMP.W	R1,R0
BT	TRGET0	The program branches to TRGET0 when R0 ≥ R1 and to TRGET1 when R0 < R1.	BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#–1,R0	T bit is not changed by ADD.	SUB.W	#1,R0
CMP/EQ	#0,R0	T bit is set when R0 = 0.	BEQ	TRGET
BT	TRGET	The program branches if R0 = 0.		

(10) Immediate Data

Byte immediate data is located in an instruction code. Word or longword immediate data is not located in instruction codes but in a memory table. The memory table is accessed by an immediate data transfer instruction (MOV) using the PC relative addressing mode with displacement.

With the SH-2A/SH2A-FPU, 17- to 28-bit immediate data can be located in an instruction code. However, for 21- to 28-bit immediate data, an OR instruction must be executed after the data is transferred to a register.

Table 2.6 Immediate Data Accessing

Classification	SH-2A/SH2A-FPU CPU		Example of Other CPU	
8-bit immediate	MOV	#H'12,R0	MOV.B	#H'12,R0
16-bit immediate	MOVI20	#H'1234,R0	MOV.W	#H'1234,R0
20-bit immediate	MOVI20	#H'12345,R0	MOV.L	#H'12345,R0
28-bit immediate	MOVI20S	#H'12345,R0	MOV.L	#H'1234567,R0
	OR	#H'67,R0		
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L	#H'12345678,R0
	.....			
	.DATA.L	H'12345678		

Note: @(disp, PC) accesses the immediate data.

(11) Absolute Address

When data is accessed by an absolute address, the absolute address value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in register indirect addressing mode.

With the SH-2A/SH2A-FPU, when data is referenced using an absolute address not exceeding 28 bits, it is also possible to transfer immediate data located in the instruction code to a register and to reference the data in register indirect addressing mode. However, when referencing data using an absolute address of 21 to 28 bits, an OR instruction must be used after the data is transferred to a register.

Table 2.7 Absolute Address Accessing

Classification	SH-2A/SH2A-FPU CPU		Example of Other CPU	
Up to 20 bits	MOVI20	#H'12345,R1	MOV.B	@H'12345,R0
	MOV.B	@R1,R0		
21 to 28 bits	MOVI20S	#H'12345,R1	MOV.B	@H'1234567,R0
	OR	#H'67,R1		
	MOV.B	@R1,R0		
29 bits or more	MOV.L	@(disp,PC),R1	MOV.B	@H'12345678,R0
	MOV.B	@R1,R0		
	.....			
	.DATA.L	H'12345678		

(12) 16-Bit/32-Bit Displacement

When data is accessed by 16-bit or 32-bit displacement, the displacement value should be placed in the memory table in advance. That value is transferred to the register by loading the immediate data during the execution of the instruction, and the data is accessed in the indexed indirect register addressing mode.

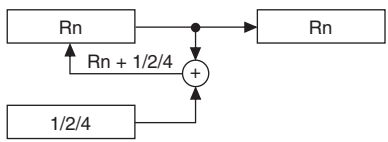
Table 2.8 Displacement Accessing

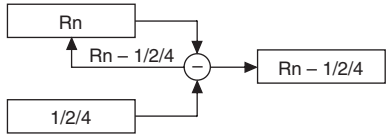
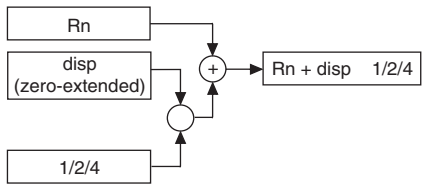
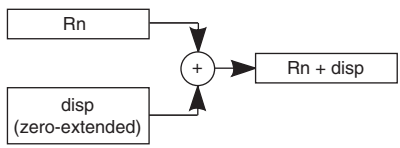
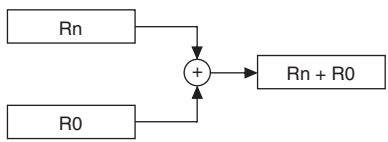
Classification	SH-2A/SH2A-FPU CPU		Example of Other CPU
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

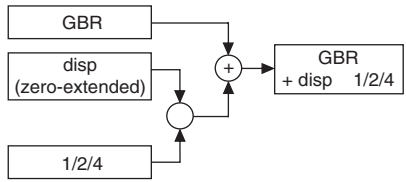
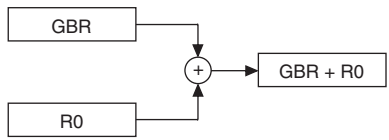
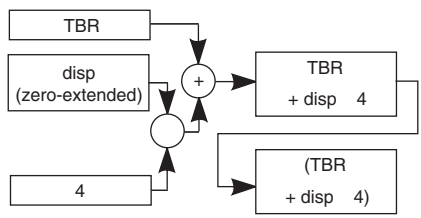
2.4.2 Addressing Modes

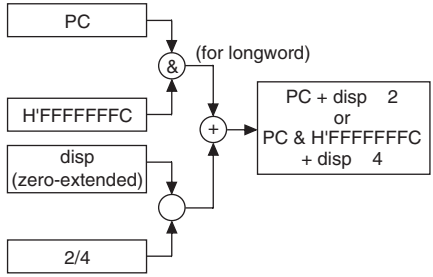
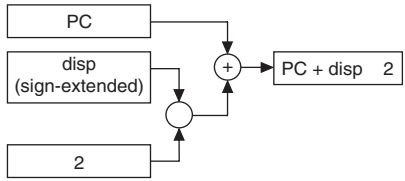
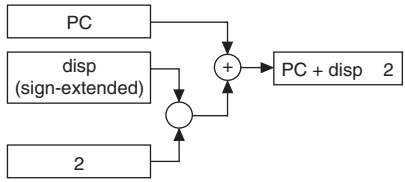
The addressing modes and effective address calculation methods are listed below.

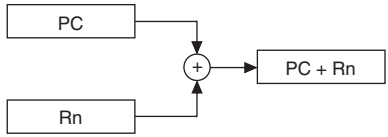
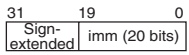
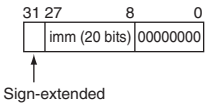
Table 2.9 Addressing Modes and Effective Addresses

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register direct	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Register indirect	@Rn	The effective address is the contents of register Rn. 	Rn
Register indirect with post-increment	@Rn+	The effective address is the contents of register Rn. A constant is added to the contents of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Rn (After instruction execution) Byte: Rn + 1 → Rn Word: Rn + 2 → Rn Longword: Rn + 4 → Rn

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Register indirect with pre-decrement	@-Rn	The effective address is the value obtained by subtracting a constant from Rn. 1 is subtracted for a byte operation, 2 for a word operation, and 4 for a longword operation. 	Byte: Rn – 1 → Rn Word: Rn – 2 → Rn Longword: Rn – 4 → Rn (Instruction is executed with Rn after this calculation)
Register indirect with displacement	@(disp:4, Rn)	The effective address is the sum of Rn and a 4-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: Rn + disp Word: Rn + disp × 2 Longword: Rn + disp × 4
Register indirect with displacement	@(disp:12 ,Rn)	The effective address is the sum of Rn and a 12-bit displacement (disp). The value of disp is zero-extended. 	Byte: Rn + disp Word: Rn + disp Longword: Rn + disp
Indexed register indirect	@(R0,Rn)	The effective address is the sum of Rn and R0. 	Rn + R0

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
GBR indirect with displacement	@(disp:8, GBR)	The effective address is the sum of GBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and remains unchanged for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: GBR + disp Word: GBR + disp × 2 Longword: GBR + disp × 4
Indexed GBR indirect	@(R0, GBR)	The effective address is the sum of GBR value and R0. 	GBR + R0
TBR duplicate indirect with displacement	@@ (disp:8, TBR)	The effective address is the sum of TBR value and an 8-bit displacement (disp). The value of disp is zero-extended, and is multiplied by 4. 	Contents of address (TBR + disp × 4)

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC indirect with displacement	@(disp:8, PC)	The effective address is the sum of PC value and an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: PC + disp × 2 Longword: PC & H'FFFFFFFC + disp × 4
PC relative	disp:8	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 8-bit displacement (disp). 	PC + disp × 2
	disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp). 	PC + disp × 2

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative	Rn	The effective address is the sum of PC value and Rn. 	PC + Rn
Immediate	#imm:20	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended. 	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, and the lower bits are padded with zero. 	—
	#imm:8	The 8-bit immediate data (imm) for the TST, AND, OR, and XOR instructions is zero-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the MOV, ADD, and CMP/EQ instructions is sign-extended.	—
	#imm:8	The 8-bit immediate data (imm) for the TRAPA instruction is zero-extended and then quadrupled.	—
	#imm:3	The 3-bit immediate data (imm) for the BAND, BOR, BXOR, BST, BLD, BSET, and BCLR instructions indicates the target bit location.	—

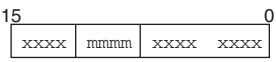
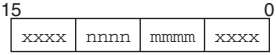
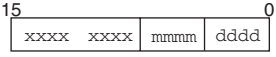
2.4.3 Instruction Format

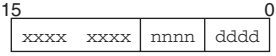
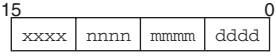
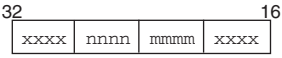
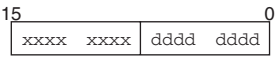
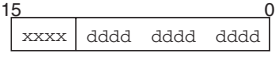
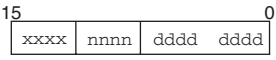
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

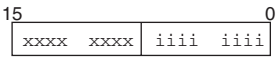
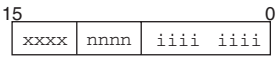
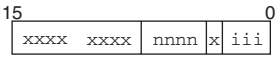
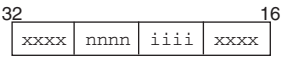
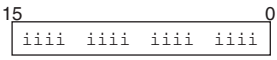
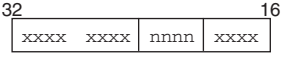
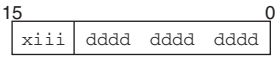
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

Table 2.10 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 15 0 xxxx xxxx xxxx xxxx	—	—	NOP
n format 15 0 xxxx nnnn xxxx xxxx	—	nnnn: Register direct	MOVT Rn
	Control register or system register	nnnn: Register direct	STS MACH,Rn
	R0 (Register direct)	nnnn: Register direct	DIVU R0,Rn
	Control register or system register	nnnn: Register indirect with pre-decrement	STC.L SR,@-Rn
	mmmm: Register direct	R15 (Register indirect with pre-decrement)	MOVMU.L Rm,@-R15
	R15 (Register indirect with post-increment)	nnnn: Register direct	MOVMU.L @R15+,Rn
	R0 (Register direct)	nnnn: (Register indirect with post-increment)	MOV.L R0,@Rn+

Instruction Formats	Source Operand	Destination Operand	Example
m format 	mmmm: Register direct	Control register or system register	LDC Rm,SR
	mmmm: Register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: Register indirect	—	JMP @Rm
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	MOV.L @-Rm,R0
	mmmm: PC relative using Rm	—	BRAF Rm
nm format 	mmmm: Register direct	nnnn: Register direct	ADD Rm,Rn
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,@Rn
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @Rm+,Rn
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm,@-Rn
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm),R0

Instruction Formats	Source Operand	Destination Operand	Example
nd4 format 	R0 (Register direct)	nnnndddd: Register indirect with displacement	MOV.B R0,@(disp,Rn)
nmd format 	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp,Rm),Rn
nmd12 format 	mmmm: Register direct	nnnndddd: Register indirect with displacement	MOV.L Rm,@(disp12,Rn)
	mmmmdddd: Register indirect with displacement	nnnn: Register direct	MOV.L @(disp12,Rm),Rn
d format 	dddddddd: GBR indirect with displacement	R0 (Register direct)	MOV.L @(disp,GBR),R0
	R0 (Register direct)	dddddddd: GBR indirect with displacement	MOV.L R0,@(disp,GBR)
	dddddddd: PC relative with displacement	R0 (Register direct)	MOVA @(disp,PC),R0
	dddddddd: TBR duplicate indirect with displacement	—	JSR/N @@(disp8,TBR)
	dddddddd: PC relative	—	BF label
d12 format 	ddddddddddddd: PC relative	—	BRA label (label = disp + PC)
nd8 format 	dddddddd: PC relative with displacement	nnnn: Register direct	MOV.L @(disp,PC),Rn

Instruction Formats	Source Operand	Destination Operand	Example
i format 	iiiiiiii: Immediate	Indexed GBR indirect	AND.B #imm,@(R0,GBR)
	iiiiiiii: Immediate	R0 (Register direct)	AND #imm,R0
	iiiiiiii: Immediate	—	TRAPA #imm
ni format 	iiiiiiii: Immediate	nnnn: Register direct	ADD #imm,Rn
ni3 format 	nnnn: Register direct iii: Immediate	—	BLD #imm3,Rn
	—	nnnn: Register direct iii: Immediate	BST #imm3,Rn
ni20 format  	iiiiiiiiiiii iiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format  	nnnndddddddd dddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12, Rn)
	—	nnnndddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12, Rn)

Note: * In multiply-and-accumulate instructions, nnnn is the source register.

2.5 Instruction Set

2.5.1 Instruction Set by Classification

Table 2.11 lists the instructions according to their classification.

Table 2.11 Classification of Instructions

Classification	Types	Operation Code	Function	No. of Instructions
Data transfer	13	MOV	Data transfer Immediate data transfer Peripheral module data transfer Structure data transfer Reverse stack transfer	62
		MOVA	Effective address transfer	
		MOVI20	20-bit immediate data transfer	
		MOVI20S	20-bit immediate data transfer 8-bit left-shit	
		MOVML	R0–Rn register save/restore	
		MOVMU	Rn–R14 and PR register save/restore	
		MOVRT	T bit inversion and transfer to Rn	
		MOVT	T bit transfer	
		MOVU	Unsigned data transfer	
		NOTT	T bit inversion	
		PREF	Prefetch to operand cache	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	

Classification	Types	Operation Code	Function	No. of Instructions
Arithmetic operations	26	ADD	Binary addition	40
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		CLIPS	Signed saturation value comparison	
		CLIPU	Unsigned saturation value comparison	
		DIVS	Signed division (32 ÷ 32)	
		DIVU	Unsigned division (32 ÷ 32)	
		DIV1	One-step division	
		DIV0S	Initialization of signed one-step division	
		DIV0U	Initialization of unsigned one-step division	
		DMULS	Signed double-precision multiplication	
		DMULU	Unsigned double-precision multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-precision multiply-and-accumulate operation	
		MUL	Double-precision multiply operation	
		MULR	Signed multiplication with result storage in Rn	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
		SUBV	Binary subtraction with underflow	

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	

Classification	Types	Operation Code	Function	No. of Instructions
Branch	10	BF	Conditional branch, conditional delayed branch (branch when T = 0)	15
		BT	Conditional branch, conditional delayed branch (branch when T = 1)	
		BRA	Unconditional delayed branch	
		BRAF	Unconditional delayed branch	
		BSR	Delayed branch to subroutine procedure	
		BSRF	Delayed branch to subroutine procedure	
		JMP	Unconditional delayed branch	
		JSR	Branch to subroutine procedure	
			Delayed branch to subroutine procedure	
		RTS	Return from subroutine procedure	
			Delayed return from subroutine procedure	
		RTV/N	Return from subroutine procedure with Rm → R0 transfer	
System control	14	CLRT	T bit clear	36
		CLRMAC	MAC register clear	
		LDBANK	Register restoration from specified register bank entry	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RESBANK	Register restoration from register bank	
		RTE	Return from exception handling	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STBANK	Register save to specified register bank entry	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	

Classification	Types	Operation Code	Function	No. of Instructions
Floating-point instructions	19	FABS	Floating-point absolute value	48
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FCNVDS	Conversion from double-precision to single-precision	
		FCNVSD	Conversion from single-precision to double -precision	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Conversion from integer to floating-point	
		FMAC	Floating-point multiply and accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	
		FSCHG	SZ bit inversion	
		FSQRT	Floating-point square root	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	

Classification	Types	Operation Code	Function	No. of Instructions
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	112			253

The table below shows the format of instruction codes, operation, and execution states. They are described by using this format according to their classification.

Instruction	Instruction Code	Operation	Execution Cycles	T Bit
Indicated by mnemonic.	Indicated in MSB ↔ LSB order.	Indicates summary of operation.	Value when no wait states are inserted.*1	Value of T bit after instruction is executed.
[Legend] OP.Sz SRC, DEST OP: Operation code Sz: Size SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	[Legend] mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement	[Legend] →, ←: Transfer direction (xx): Memory operand M/Q/T: Flag bits in SR &: Logical AND of each bit |: Logical OR of each bit ^: Exclusive logical OR of each bit ~: Logical NOT of each bit <<n: n-bit left shift >>n: n-bit right shift		[Legend] —: No change

Notes: 1. Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states will be increased in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

2. Depending on the operand size, displacement is scaled by ×1, ×2, or ×4. For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.5.2 Data Transfer Instructions

Table 2.12 Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/SH2A-FPU
MOV	#imm, Rn	1110nnnniiiiiiii	imm → sign extension → Rn	1	—	Yes	Yes	
MOV.W	@(disp, PC),Rn	1001nnnnddddddd	(disp × 2 + PC) → sign extension → Rn	1	—	Yes	Yes	
MOV.L	@(disp, PC),Rn	1101nnnnddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Yes	
MOV	Rm, Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Yes	
MOV.B	Rm, @Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Yes	
MOV.W	Rm, @Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Yes	
MOV.L	Rm, @Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Yes	
MOV.B	@Rm, Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.W	@Rm, Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.L	@Rm, Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Yes	
MOV.B	Rm, @-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Yes	
MOV.W	Rm, @-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Yes	
MOV.L	Rm, @-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Yes	
MOV.B	@Rm+, Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Yes	
MOV.W	@Rm+, Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Yes	
MOV.L	@Rm+, Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Yes	
MOV.B	R0, @(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Yes	
MOV.W	R0, @(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Yes	
MOV.L	Rm, @(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Yes	
MOV.B	@(disp, Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
MOV.W	@(disp, Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Yes	
MOV.L	@(disp, Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Yes	
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Yes	
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—	Yes	Yes	
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—	Yes	Yes	
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Yes	
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Yes	
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Yes	
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Yes	
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Yes	
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	Yes	
MOV.B	R0,@Rn+	0100nnnn10001011	R0 → (Rn), Rn + 1 → Rn	1	—			Yes
MOV.W	R0,@Rn+	0100nnnn10011011	R0 → (Rn), Rn + 2 → Rn	1	—			Yes
MOV.L	R0,@Rn+	0100nnnn10101011	R0 → Rn), Rn + 4 → Rn	1	—			Yes
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2 → Rm, (Rm) → sign extension → R0	1	—			Yes
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4 → Rm, (Rm) → R0	1	—			Yes
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm → (disp + Rn)	1	—			Yes
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm → (disp × 2 + Rn)	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm → (disp × 4 + Rn)	1	—			Yes
MOV.B	@(disp12, Rm), Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp + Rm) → sign extension → Rn	1	—			Yes
MOV.W	@(disp12, Rm), Rn	0011nnnnmmmm0001 0101dddddddddddd	(disp × 2 + Rm) → sign extension → Rn	1	—			Yes
MOV.L	@(disp12, Rm), Rn	0011nnnnmmmm0001 0110dddddddddddd	(disp × 4 + Rm) → Rn	1	—			Yes
MOVA	@(disp,PC),R0	11000111dddddddd	disp × 4 + PC → R0	1	—	Yes	Yes	
MOVI20	#imm20, Rn	0000nnnniiii0000 iiiiiiiiiiiiiiii	imm → sign extension → Rn	1	—			Yes
MOVI20S	#imm20, Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm << 8 → sign extension → Rn	1	—			Yes
MOVML.L	Rm, @-R15	0100mmmm11110001	R15-4 → R15, Rm → (R15) R15-4 → R15, Rm-1 → (R15) : R15-4 → R15, R0 → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVML.L	@R15+, Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rm as PR	1 to 16	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
MOVMU.L	Rm, @-R15	0100mmmm11110000	R15-4 → R15, PR → (R15) R15-4 → R15, R14 → (R15) : R15-4 → R15, Rm → (R15) Note: When Rm = R15, read Rm as PR	1 to 16	—			Yes
MOVMU.L	@R15+, Rn	0100nnnn11110100	(R15) → Rn, R15 + 4 → R15 (R15) → Rn + 1, R15 + 4 → R15 : (R15) → R14, R15 + 4 → R15 (R15) → PR Note: When Rn = R15, read Rm as PR	1 to 16	—			Yes
MOVRT	Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT	Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	
MOVU.B	@(disp12, Rm), Rn	0011nnnnmmmm0001 1000ddddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12, Rm), Rn	0011nnnnmmmm0001 1001ddddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes
NOTT		0000000001101000	~T → T	1	Operation result			Yes
PREF	@Rn	0000nnnn10000011	(Rn) → operand cache	1	—		Yes	
SWAP.B	Rm, Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—	Yes	Yes	
SWAP.W	Rm, Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	Yes	
XTRCT	Rm, Rn	0010nnnnmmmm1101	Middle 32 bits of Rm:Rn → Rn	1	—	Yes	Yes	

2.5.3 Arithmetic Operation Instructions

Table 2.13 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
ADD	Rm, Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—	Yes	Yes	
ADD	#imm, Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—	Yes	Yes	
ADDC	Rm, Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, carry → T	1	Carry	Yes	Yes	
ADDV	Rm, Rn	0011nnnnmmmm1111	Rn + Rm → Rn, overflow → T	1	Over-flow	Yes	Yes	
CMP/EQ	#imm, R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/EQ	Rm, Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/GE	Rm, Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/HI	Rm, Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	

Instruction	Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
CMP/STR Rm, Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CLIPS.B Rn	0100nnnn10010001	When Rn > (H'0000007F), (H'0000007F) → Rn, 1 → CS when Rn < (H'FFFFFF80), (H'FFFFFF80) → Rn, 1 → CS	1	—			Yes
CLIPS.W Rn	0100nnnn10010101	When Rn > (H'00007FFF), (H'00007FFF) → Rn, 1 → CS When Rn < (H'FFFF8000), (H'FFFF8000) → Rn, 1 → CS	1	—			Yes
CLIPU.B Rn	0100nnnn10000001	When Rn > (H'000000FF), (H'000000FF) → Rn, 1 → CS	1	—			Yes
CLIPU.W Rn	0100nnnn10000101	When Rn > (H'0000FFFF), (H'0000FFFF) → Rn, 1 → CS	1	—			Yes
DIV1 Rm, Rn	0011nnnnmmmm0100	1-step division (Rn ÷ Rm)	1	Calculation result	Yes	Yes	
DIV0S Rm, Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result	Yes	Yes	
DIV0U	0000000000011001	0 → M/Q/T	1	0	Yes	Yes	
DIVS R0, Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
DIVU	R0, Rn	0100nnnn10000100	Unsigned operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	34	—			Yes
DMULS.L	Rm, Rn	0011nnnnmmmm1101	Signed operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	
DMULU.L	Rm, Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	
DT	Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Comparison result	Yes	Yes	
EXTS.B	Rm, Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes	Yes	
EXTS.W	Rm, Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes	Yes	
EXTU.B	Rm, Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	
EXTU.W	Rm, Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—	Yes	Yes	
MAC.L	@Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	4	—	Yes	Yes	
MAC.W	@Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3	—	Yes	Yes	
MUL.L	Rm, Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	Yes	
MULR	R0, Rn	0100nnnn10000000	R0 × Rn → Rn 32 × 32 → 32 bits	2				Yes
MULS.W	Rm, Rn	0010nnnnmmmm1111	Signed operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
MULU.W	Rm, Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	
NEG	Rm, Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Yes	
NEGC	Rm, Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	
SUB	Rm, Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Yes	
SUBC	Rm, Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	
SUBV	Rm, Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Over-flow	Yes	Yes	

2.5.4 Logic Operation Instructions

Table 2.14 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
AND	Rm, Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	
AND	#imm, R0	11001001iiiiiiii	R0 & imm → R0	1	—	Yes	Yes	
AND.B	#imm, @(R0, GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—	Yes	Yes	
NOT	Rm, Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Yes	
OR	Rm, Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Yes	
OR	#imm, R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Yes	
OR.B	#imm, @(R0, GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Yes	
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
TST	Rm, Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	
TST	#imm, R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	
TST.B	#imm, @(R0, GBR)	11001100iiiiiiii	(R0 + GBR) & imm When the result is 0, 1 → T Otherwise, 0 → T	3	Test result	Yes	Yes	
XOR	Rm, Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	
XOR	#imm, R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Yes	
XOR.B	#imm, @(R0, GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Yes	

2.5.5 Shift Instructions

Table 2.15 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB	Yes	Yes	
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB	Yes	Yes	
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB	Yes	Yes	
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Yes	
SHAD	Rm, Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB	Yes	Yes	
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB	Yes	Yes	
SHLD	Rm, Rn	0100nnnnmmmm1101	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Yes	
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB	Yes	Yes	
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB	Yes	Yes	
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—	Yes	Yes	
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—	Yes	Yes	
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—	Yes	Yes	
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—	Yes	Yes	
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—	Yes	Yes	
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—	Yes	Yes	

2.5.6 Branch Instructions

Table 2.16 Branch Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
BF	label	10001011dddddddd	When T = 0, disp × 2 + PC → PC, When T = 1, nop	3/1*	—	Yes	Yes	
BF/S	label	10001111dddddddd	Delayed branch When T = 0, disp × 2 + PC → PC, When T = 1, nop	2/1*	—	Yes	Yes	
BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Yes	
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Yes	
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Yes	
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Yes	
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Yes	
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Yes	
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	Yes	
JSR/N	@Rm	0100mmmm01001011	PC-2 → PR, Rm → PC	3	—			Yes
JSR/N	@@(disp8, TBR)	10000011dddddddd	PC-2 → PR, (disp × 4 + TBR) → PC	5	—			Yes
RTS		0000000000001011	Delayed branch, PR → PC	2	—	Yes	Yes	
RTS/N		0000000001101011	PR → PC	3	—			Yes
RTV/N	Rm	0000mmmm01111011	Rm → R0, PR → PC	3	—			Yes

Note: * One cycle when the program does not branch.

2.5.7 System Control Instructions

Table 2.17 System Control Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
CLRT		0000000000001000	0 → T	1	0	Yes	Yes	
CLRMAC		0000000000101000	0 → MACH,MACL	1	—	Yes	Yes	
LDBANK	@Rm,R0	0100mmmm11100101	(Specified register bank entry) → R0	6	—			Yes
LDC	Rm,SR	0100mmmm00001110	Rm → SR	3	LSB	Yes	Yes	
LDC	Rm,TBR	0100mmmm01001010	Rm → TBR	1	—			Yes
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—	Yes	Yes	
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—	Yes	Yes	
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Yes	
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Yes	
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Yes	
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—	Yes	Yes	
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—	Yes	Yes	
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—	Yes	Yes	
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Yes	
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Yes	
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Yes	
NOP		0000000000001001	No operation	1	—	Yes	Yes	
RESBANK		0000000001011011	Bank → R0 to R14, GBR, MACH, MACL, PR	9*	—			Yes
RTE		0000000000101011	Delayed branch, stack area → PC/SR	6	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
SETT		0000000000011000	1 → T	1	1	Yes	Yes	
SLEEP		0000000000011011	Sleep	5	—	Yes	Yes	
STBANK	R0,@Rn	0100nnnn11100001	R0 → (specified register bank entry)	7	—			Yes
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Yes	
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—			Yes
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—	Yes	Yes	
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—	Yes	Yes	
STC.L	SR,@-Rn	0100nnnn00000011	Rn-4 → Rn, SR → (Rn)	2	—	Yes	Yes	
STC.L	GBR,@-Rn	0100nnnn00010011	Rn-4 → Rn, GBR → (Rn)	1	—	Yes	Yes	
STC.L	VBR,@-Rn	0100nnnn00100011	Rn-4 → Rn, VBR → (Rn)	1	—	Yes	Yes	
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—	Yes	Yes	
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—	Yes	Yes	
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—	Yes	Yes	
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Yes	
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Yes	
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Yes	
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Yes	

Notes: Instruction execution cycles: The execution cycles shown in the table are minimums. In practice, the number of instruction execution states in cases such as the following:

a. When there is a conflict between an instruction fetch and a data access

b. When the destination register of a load instruction (memory → register) is the same as the register used by the next instruction.

* In the event of bank overflow, the number of cycles is 19.

2.5.8 Floating-Point Operation Instructions

Table 2.18 Floating-Point Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2E	SH4	SH-2A/ SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn| → FRn	1	—	Yes	Yes	
FABS	DRn	1111nnn001011101	|DRn| → DRn	1	—		Yes	
FADD	FRm, FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—	Yes	Yes	
FADD	DRm, DRn	1111nnn0mmm00000	DRn + DRm → DRn	6	—		Yes	
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result	Yes	Yes	
FCMP/EQ	DRm, DRn	1111nnn0mmm00100	(DRn = DRm)? 1:0 → T	2	Comparison result		Yes	
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Compa-rison result	Yes	Yes	
FCMP/GT	DRm, DRn	1111nnn0mmm00101	(DRn > DRm)? 1:0 → T	2	Comparison result		Yes	
FCNVDS	DRm, FPUL	1111mmm010111101	(float) DRm → FPUL	2	—		Yes	
FCNVSD	FPUL, DRn	1111nnn010101101	(double) FPUL → DRn	2	—		Yes	
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm → FRn	10	—	Yes	Yes	
FDIV	DRm, DRn	1111nnn0mmm00011	DRn/DRm → DRn	23	—		Yes	
FLDI0	FRn	1111nnnn10001101	0 × 00000000 → FRn	1	—	Yes	Yes	
FLDI1	FRn	1111nnnn10011101	0 × 3F800000 → FRn	1	—	Yes	Yes	
FLDS	FRm, FPUL	1111mmmm00011101	FRm → FPUL	1	—	Yes	Yes	
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL → FRn	1	—	Yes	Yes	
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL → DRn	2	—		Yes	
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0 × FRm+FRn → FRn	1	—	Yes	Yes	
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—	Yes	Yes	
FMOV	DRm, DRn	1111nnn0mmm01100	DRm → DRn	2	—		Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	Compatibility SH4	Compatibility SH-2A/SH2A-FPU
FMOV.S	@(R0, Rm), FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—	Yes	Yes	
FMOV.D	@(R0, Rm), DRn	1111nnn0mmmm0110	(R0 + Rm) → DRn	2	—		Yes	
FMOV.S	@Rm+, FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+=4	1	—	Yes	Yes	
FMOV.D	@Rm+, DRn	1111nnn0mmmm1001	(Rm) → DRn, Rm += 8	2	—		Yes	
FMOV.S	@Rm, FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—	Yes	Yes	
FMOV.D	@Rm, DRn	1111nnn0mmmm1000	(Rm) → DRn	2	—		Yes	
FMOV.S	@(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp × 4 + Rm) → FRn	1	—			Yes
FMOV.D	@(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp × 8 + Rm) → DRn	2	—			Yes
FMOV.S	FRm, @(R0,Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—	Yes	Yes	
FMOV.D	DRm, @(R0,Rn)	1111nnnnmmm00111	DRm → (R0 + Rn)	2	—		Yes	
FMOV.S	FRm, @-Rn	1111nnnnmmmm1011	Rn-=4, FRm → (Rn)	1	—	Yes	Yes	
FMOV.D	DRm, @-Rn	1111nnnnmmm01011	Rn-=8, DRm → (Rn)	2	—		Yes	
FMOV.S	FRm, @Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—	Yes	Yes	
FMOV.D	DRm, @Rn	1111nnnnmmm01010	DRm → (Rn)	2	—		Yes	
FMOV.S	FRm, @(disp12,Rn)	0011nnnnmmmm0001 0011dddddddddddd	FRm → (disp × 4 + Rn)	1	—			Yes
FMOV.D	DRm, @(disp12,Rn)	0011nnnnmmm00001 0011dddddddddddd	DRm → (disp × 8 + Rn)	2	—			Yes
FMUL	FRm, FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—	Yes	Yes	
FMUL	DRm, DRn	1111nnn0mmm00010	DRn × DRm → DRn	6	—		Yes	
FNEG	FRn	1111nnnn01001101	-FRn → FRn	1	—	Yes	Yes	
FNEG	DRn	1111nnn001001101	-DRn → DRn	1	—		Yes	
FSCHG		1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		Yes	
FSQRT	FRn	1111nnnn01101101	√FRn → FRn	9	—		Yes	
FSQRT	DRn	1111nnn001101101	√DRn → DRn	22	—		Yes	
FSTS	FPUL,FRn	1111nnnn00001101	FPUL → FRn	1	—	Yes	Yes	
FSUB	FRm, FRn	1111nnnnmmmm0001	FRn-FRm → FRn	1	—	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FSUB	DRm, DRn	1111nnn0mmm00001	DRn-DRm → DRn	6	—		Yes	
FTRC	FRm, FPUL	1111mmmm00111101	(long)FRm → FPUL	1	—	Yes	Yes	
FTRC	DRm, FPUL	1111mmm000111101	(long)DRm → FPUL	2	—		Yes	

2.5.9 FPU-Related CPU Instructions

Table 2.19 FPU-Related CPU Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
LDS	Rm,FPSCR	0100mmmm01101010	Rm → FPSCR	1	—	Yes	Yes	
LDS	Rm,FPUL	0100mmmm01011010	Rm → FPUL	1	—	Yes	Yes	
LDS.L	@Rm+, FPSCR	0100mmmm01100110	(Rm) → FPSCR, Rm+=4	1	—	Yes	Yes	
LDS.L	@Rm+, FPUL	0100mmmm01010110	(Rm) → FPUL, Rm+=4	1	—	Yes	Yes	
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—	Yes	Yes	
STS	FPUL,Rn	0000nnnn01011010	FPUL → Rn	1	—	Yes	Yes	
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn-=4, FPCSR → (Rn)	1	—	Yes	Yes	
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn-=4, FPUL → (Rn)	1	—	Yes	Yes	

2.5.10 Bit Manipulation Instructions

Table 2.20 Bit Manipulation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp + Rn)) & T → T	3	Operation result			Yes

Instruction	Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
BANDNOT.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of (disp + Rn)) & T → T	3	Operation result			Yes
BCLR.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0000dddddddddddd	0 → (imm of (disp + Rn))	3	—			Yes
BCLR #imm3,Rn	10000110nnnn0iii	0 → imm of Rn	1	—			Yes
BLD.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0011dddddddddddd	(imm of (disp + Rn)) →	3	Operation result			Yes
BLD #imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result			Yes
BLDNOT.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result			Yes
BOR.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BORNOT.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BSET.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1 → (imm of (disp + Rn))	3	—			Yes
BSET #imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—			Yes
BST.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T → (imm of (disp + Rn))	3	—			Yes
BST #imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—			Yes
BXOR.B #imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

2.6 Processing States

The CPU has four processing states: reset, exception handling, program execution, and power-down. Figure 2.8 shows the transitions between the states.

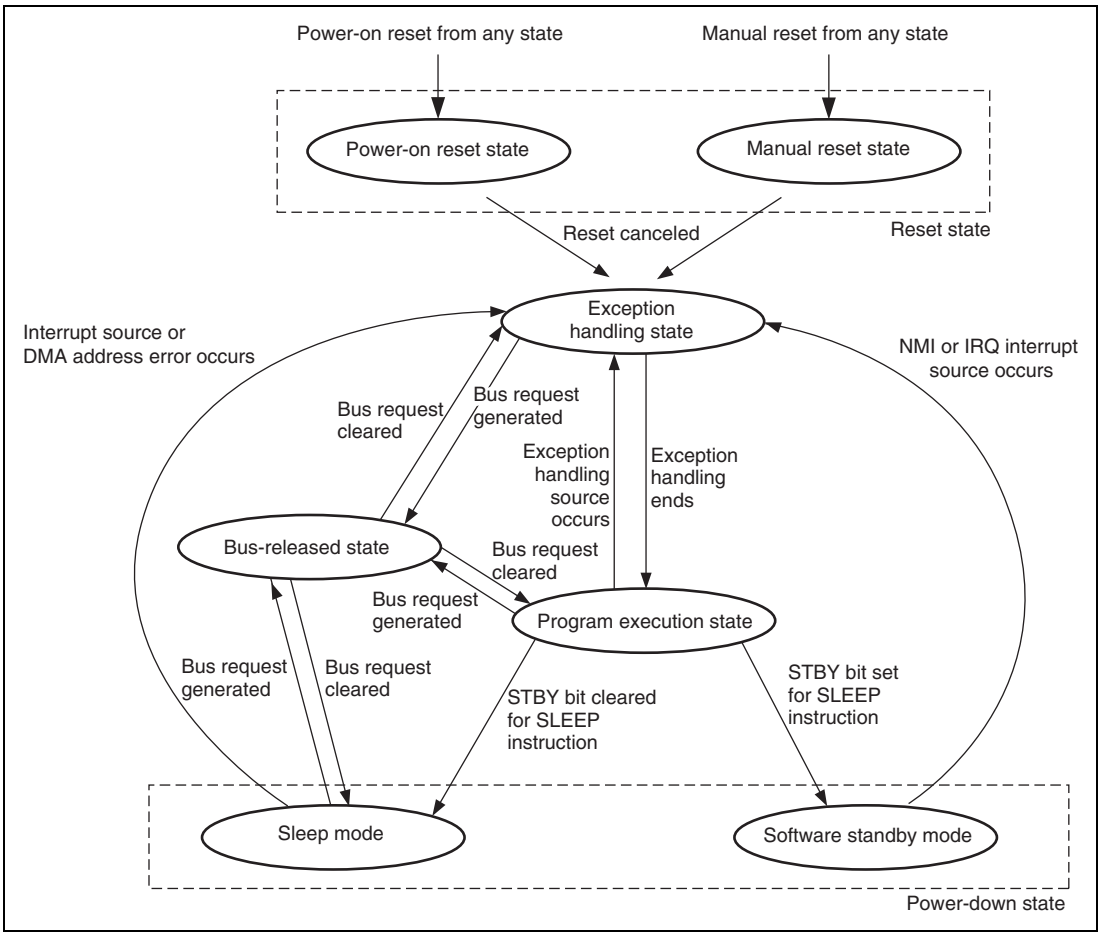


Figure 2.8 Transitions between Processing States

(1) Reset State

In this state, the CPU is reset. There are two kinds of reset, power-on reset and manual reset.

(2) Exception Handling State

The exception handling state is a transient state that occurs when exception handling sources such as resets or interrupts alters the CPU's processing state flow.

For a reset, the initial values of the program counter (PC) (execution start address) and stack pointer (SP) are fetched from the exception handling vector table and stored; the CPU then branches to the execution start address and execution of the program begins.

For an interrupt, the stack pointer (SP) is accessed and the program counter (PC) and status register (SR) are saved to the stack area. The exception service routine start address is fetched from the exception handling vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

(3) Program Execution State

In the program execution state, the CPU sequentially executes the program.

(4) Power-Down State

In the power-down state, the CPU stops operating to reduce power consumption. The SLEEP instruction places the CPU in the sleep mode or the software standby mode.

Section 3 MCU Operating Modes

3.1 Selection of Operating Modes

This LSI has four MCU operating modes and three on-chip flash memory programming modes. The operating mode is determined by the setting of FWE, MD1, and MD0 pins. Table 3.1 shows the allowable combinations of these pin settings; do not set these pins in the other way than the shown combinations.

When power is applied to the system, be sure to conduct power-on reset.

The MCU operating mode can be selected from MCU extension modes 0 to 2 and single chip mode. For the on-chip flash memory programming mode, boot mode, user boot mode, and user program mode which are on-chip programming modes are available.

Table 3.1 Selection of Operating Modes

	Pin Setting					
Mode No.	**FWE**	**MD1**	**MD0**	**Mode Name**	**On-Chip ROM**	**Bus Width of CS0 Space**
Mode 0	0	0	0	MCU extension mode 0	Not active	32
Mode 1	0	0	1	MCU extension mode 1	Not active	16
Mode 2	0	1	0	MCU extension mode 2	Active	Set by CS0BCR in BSC
Mode 3	0	1	1	Single chip mode	Active	—
Mode 4*[1]	1	0	0	Boot mode	Active	Set by CS0BCR in BSC
Mode 5*[1]	1	0	1	User boot mode	Active	Set by CS0BCR in BSC
Mode 6*[1]	1	1	0	User program mode	Active	Set by CS0BCR in BSC
Mode 7*[1]*[2]	1	1	1	USB boot mode	Active	—
Mode 7*[1]*[3]	1	1	1	User program mode	Active	—

Notes: 1. Flash memory programming mode.
2. When always FWE = 1, after the power has been on.
3. If FWE = 0 when power-on reset has been released, and if FWE = 1 when the MCU operation has been set, transition to the user program mode is executed in a single chip state.

3.2 Input/Output Pins

Table 3.2 describes the configuration of operating mode related pin.

Table 3.2 Pin Configuration

Pin Name	Input/Output	Function
MD0	Input	Designates operating mode through the level applied to this pin
MD1	Input	Designates operating mode through the level applied to this pin
FWE	Input	Enables, by hardware, programming/erasing of the on-chip flash memory

3.3 Operating Modes

3.3.1 Mode 0 (MCU Extension Mode 0)

In this mode, CS0 space becomes external memory spaces with 32-bit bus width.

3.3.2 Mode 1 (MCU Extension Mode 1)

In this mode, CS0 space becomes external memory spaces with 16-bit bus width.

3.3.3 Mode 2 (MCU Extension Mode 2)

The on-chip ROM is active and CS0 space can be used in this mode.

3.3.4 Mode 3 (Single Chip Mode)

All ports can be used in this mode, however the external address cannot be used.

3.4 Address Map

The address map for the operating modes is shown in figures 3.1 to 3.3.

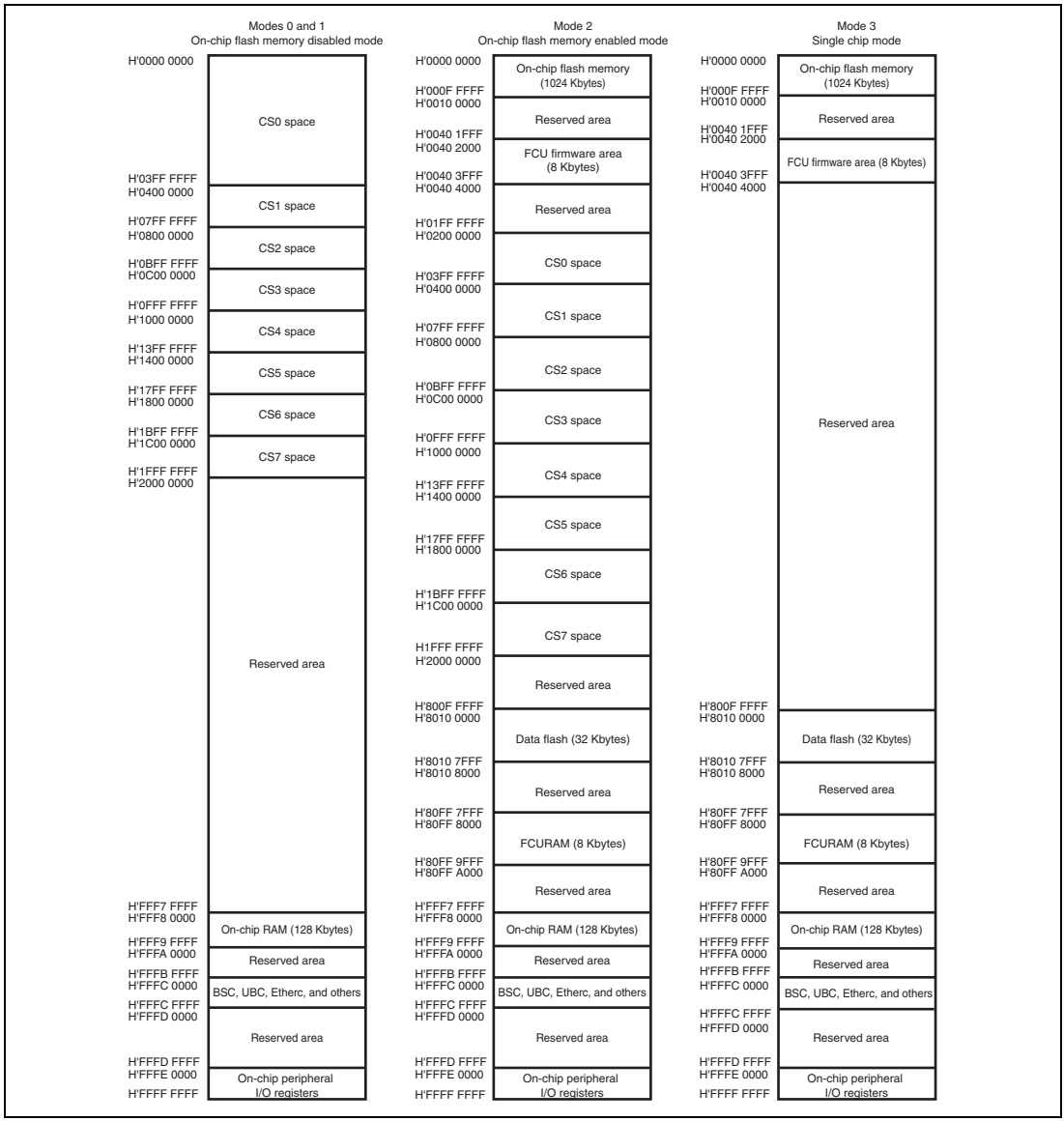


Figure 3.1 Address Map (1-Mbyte Version)

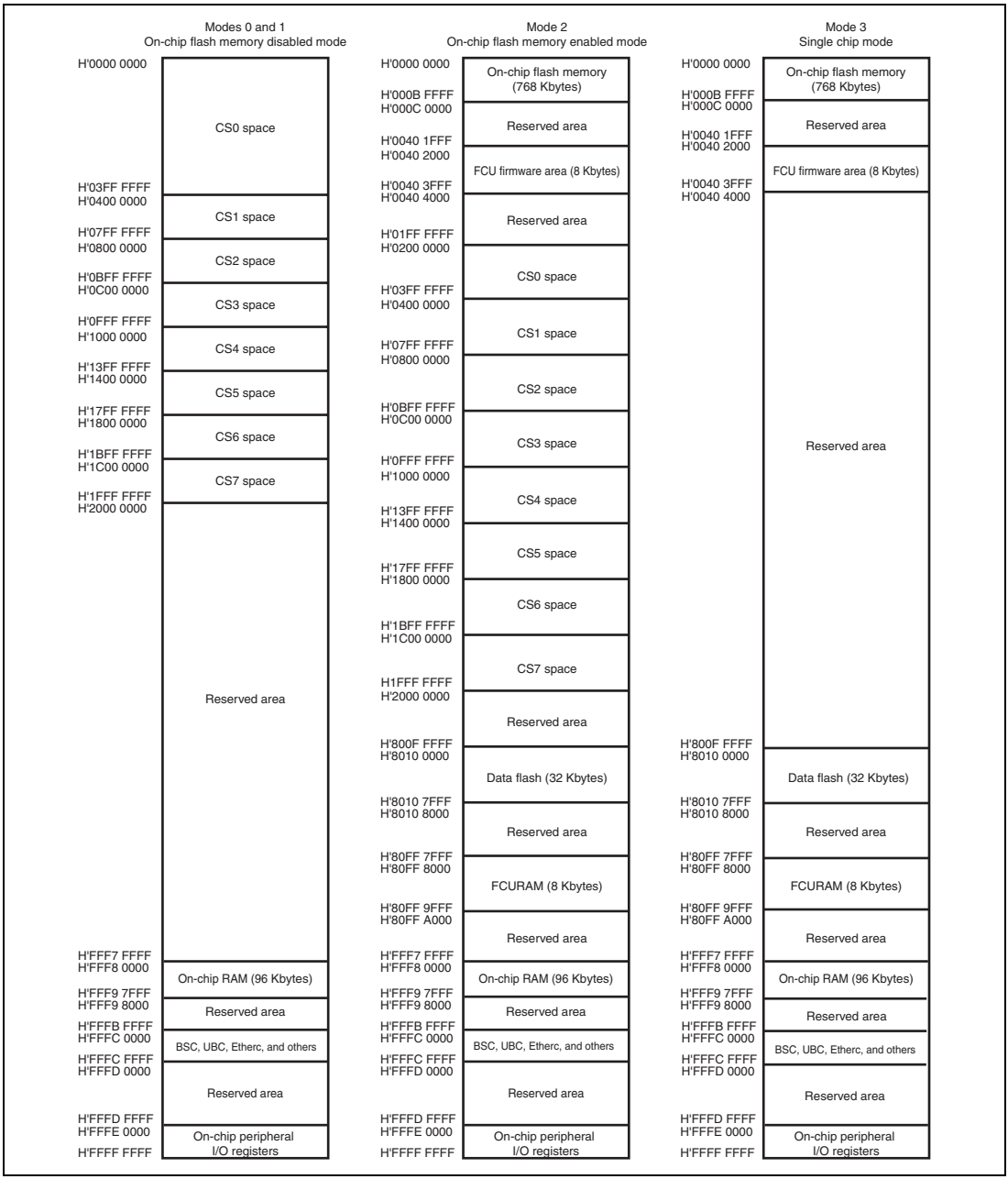


Figure 3.2 Address Map (768-Kbyte Version)

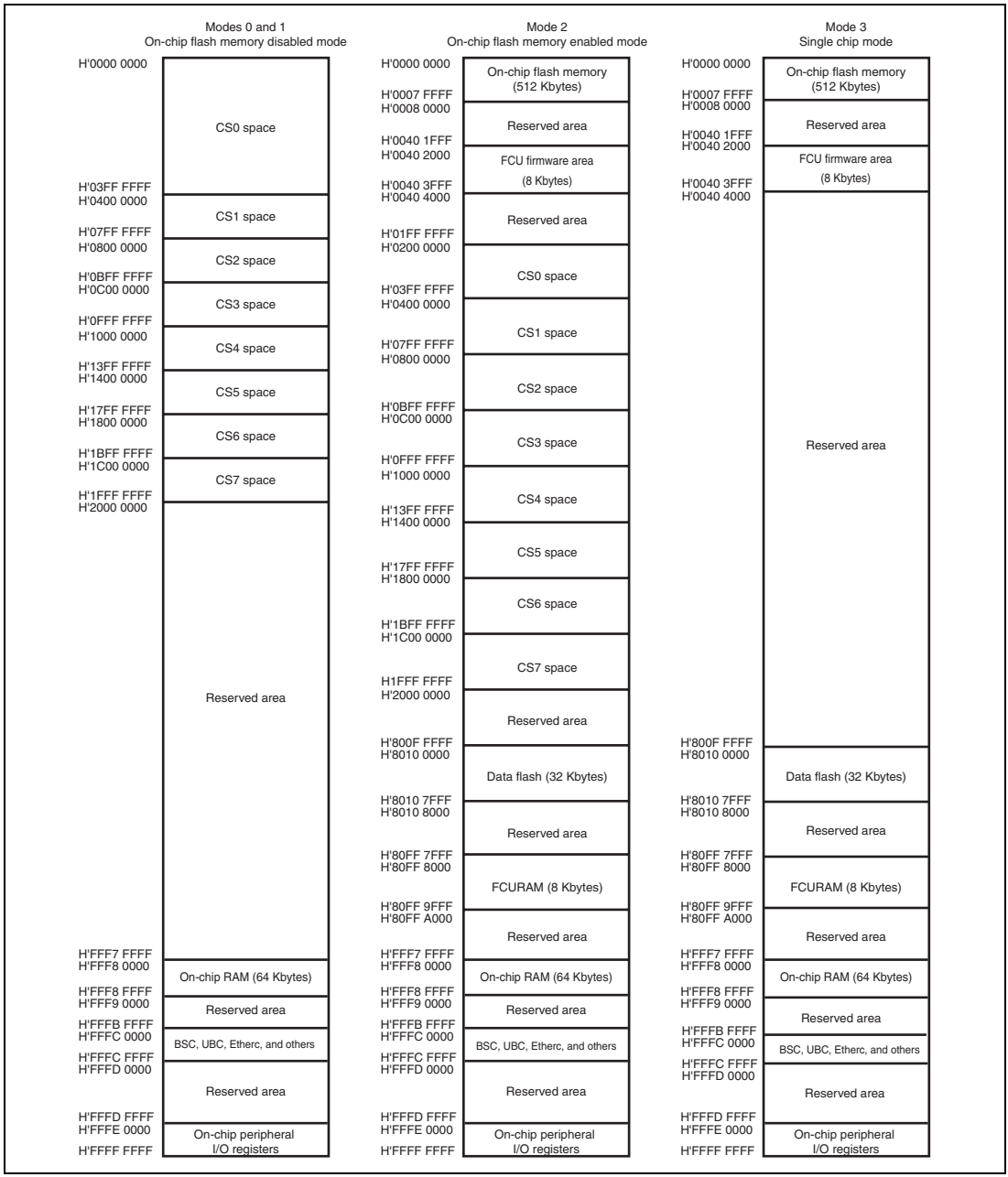


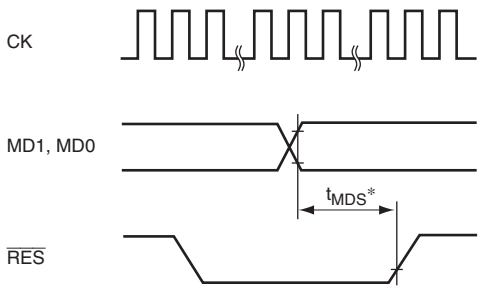
Figure 3.3 Address Map (512-Kbyte Version)

3.5 Initial State in This LSI

In the initial state of this LSI, some of on-chip modules are set in module standby state for saving power. When operating these modules, clear module standby state according to the procedure in section 30, Power-Down Modes.

3.6 Note on Changing Operating Mode

When changing operating mode while power is applied to this LSI, make sure to do it in the power-on reset state (that is, the low level is applied to the $\overline{\text{RES}}$ pin).



Note: * See section 33.3.2, Control Signal Timing.

Figure 3.4 Reset Input Timing when Changing Operating Mode

Section 4 Clock Pulse Generator (CPG)

This LSI has a clock pulse generator (CPG) that generates an internal clock (Iϕ), a peripheral clock (Pϕ), a bus clock (Bϕ), an MTU2S clock (Mϕ), and an AD clock (Aϕ). The CPG consists of a crystal oscillator, a PLL circuit, and a divider circuit.

4.1 Features

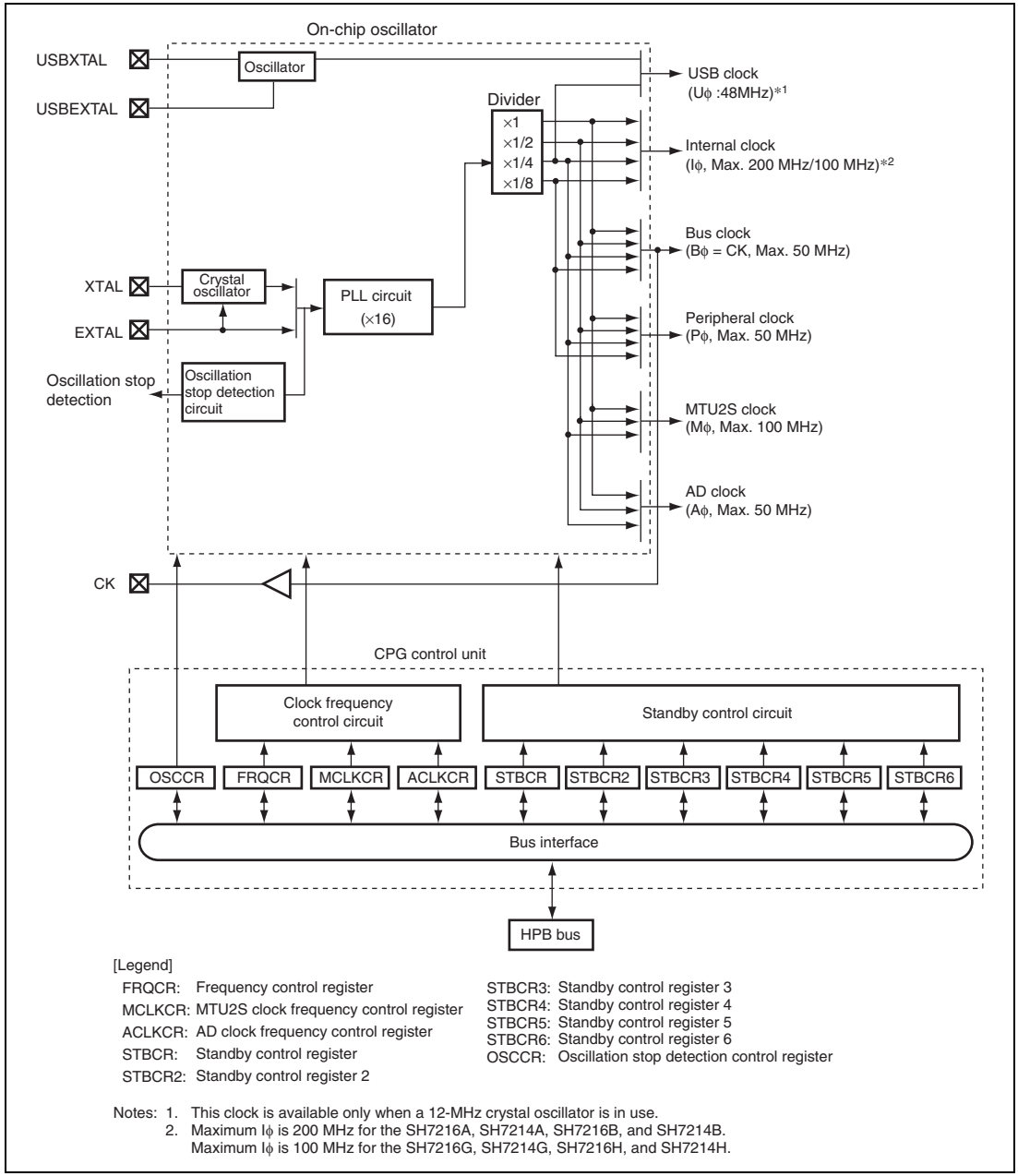
- Five clocks generated independently

  An internal clock (Iϕ) for the CPU and cache, a peripheral clock (Pϕ) for the peripheral modules, a bus clock (Bϕ = CK) for the external bus interface, an MTU2S clock (Mϕ) for the MTU2S module, and an AD clock (Aϕ) for the ADC module can be generated independently.
- Frequency change function

  Internal and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuit and divider circuit within the CPG. Frequencies are changed by software using frequency control register (FRQCR) settings.
- Power-down mode control

  The clock can be stopped for sleep mode and software standby mode, and specific modules can be stopped using the module standby function. For details on clock control in the power-down modes, see section 30, Power-Down Modes.

Figure 4.1 shows a block diagram of the clock pulse generator.



[Legend]

FRQCR: Frequency control register
MCLKCR: MTU2S clock frequency control register
ACLKCR: AD clock frequency control register
STBCR: Standby control register
STBCR2: Standby control register 2
STBCR3: Standby control register 3
STBCR4: Standby control register 4
STBCR5: Standby control register 5
STBCR6: Standby control register 6
OSCCR: Oscillation stop detection control register

Notes: 1. This clock is available only when a 12-MHz crystal oscillator is in use.
2. Maximum Iϕ is 200 MHz for the SH7216A, SH7214A, SH7216B, and SH7214B.
Maximum Iϕ is 100 MHz for the SH7216G, SH7214G, SH7216H, and SH7214H.

Figure 4.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

(1) PLL Circuit

The PLL circuit multiplies the input clock frequency from the crystal oscillator or EXTAL pin by 16.

(2) Crystal Oscillator

The crystal oscillator is an oscillation circuit in which a crystal resonator is connected to the XTAL pin or EXTAL pin. This can be used according to the clock operating mode.

(3) Divider

The divider generates a clock signal at the operating frequency used by the internal clock (Iϕ), bus clock (Bϕ), peripheral clock (Pϕ), MTU2S clock (Mϕ), or AD clock (Aϕ). The operating frequency can be 1, 1/2, 1/4, or 1/8 times the output frequency of the PLL circuit. The division ratio is set in the frequency control register (FRQCR). USB clock (Uϕ) is set as fixed 1/4 and when generating USB clock with a divider, set the crystal resonator to 12 MHz.

(4) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the frequency control register (FRQCR).

(5) Standby Control Circuit

The standby control circuit controls the states of the clock pulse generator and other modules during clock switching, or sleep or software standby mode.

(6) Frequency Control Register (FRQCR)

The frequency control register (FRQCR) has control bits assigned for the following functions: the frequency division ratios of the internal clock (Iϕ), bus clock (Bϕ), and peripheral clock (Pϕ).

(7) MTU2S Clock Frequency Control Register (MCLKCR)

The MTU2S clock frequency control register (MCLKCR) has control bits assigned for the following function: the frequency division ratio of the MTU2S clock (Mϕ).

(8) AD Clock Frequency Control Register (ACLKCR)

The AD clock frequency control register (ACLKCR) has control bits assigned for the following functions: the frequency division ratio of the AD clock (Aϕ).

(9) Standby Control Register

The standby control register has bits for controlling the power-down modes and for selecting the USB clock. See section 30, Power-Down Modes, for more information.

(10) Oscillation Stop Detection Control Register (OSCCR)

The oscillation stop detection control register (OSCCR) has an oscillation stop detection flag and a bit for selecting flag status output through an external pin.

(11) USB-only oscillator

The oscillator for USB clock only that is connected to the resonator of 48 MHz.

4.2 Input/Output Pins

Table 4.1 lists the clock pulse generator pins and their functions.

Table 4.1 Pin Configuration and Functions of the Clock Pulse Generator

Pin Name	Symbol	I/O	Function
Crystal input/output pins (clock input pins)	XTAL	Output	Connected to the crystal resonator. (Leave this pin open when the crystal resonator is not in use.)
	EXTAL	Input	Connected to the crystal resonator or used to input an external clock.
Clock output pin	CK	Output	Clock output pin. This pin can be placed in high-impedance state.
Crystal input/output pins for USB (clock input pins)	USBXTAL	Output	Connected to the crystal resonator for USB (equivalent for CSTCZ48M0X11R). Leave this pin open when the crystal resonator is not in use.
	USBEXTAL	Input	Connected to the crystal resonator for USB (equivalent for CSTCZ48M0X11R). Connect this pin to Vss when the crystal resonator is not in use.

To use the clock output (CK) pin, appropriate settings may be needed in the pin function controller (PFC) in some cases. For details, refer to section 22, Pin Function Controller (PFC).

4.3 Clock Operating Modes

Table 4.2 shows the clock operating modes of this LSI.

Table 4.2 Clock Operating Modes

	Clock I/O			
Mode	**Source**	**Output**	**PLL Circuit**	**Input to Divider**
1	EXTAL input or crystal resonator	CK*	On (× 16)	× 16

Note: * To output the clock through the CK pin, appropriate settings should be made in the PFC. For details, refer to section 22, Pin Function Controller (PFC).

The frequency of the external clock input from the EXTAL pin is multiplied by 16 in the PLL circuit before it is supplied to the on-chip modules in this LSI, which eliminates the need to generate a high-frequency clock outside the LSI. Since the input clock frequency ranging from 10 MHz to 12.5 MHz can be used, the internal clock (Iϕ) frequency ranges from 20 MHz to 200 MHz or 100 MHz.

Maximum operating frequencies*:
Iϕ = 200 MHz/100 MHz, Bϕ = 50 MHz, Pϕ = 50 MHz, Mϕ = 100 MHz, Aϕ = 50 MHz

Table 4.3 shows an example of a range for the frequency division ratios that can be specified with FRQCR.

Note: * The 200-MHz Iϕ applies to the SH7216A, SH7214A, SH7216B, and SH7214B.
The 100-MHz Iϕ applies to the SH7216G, SH7214G, SH7216H, and SH7214H.

Table 4.3 Example of Relationship between Clock Operating Mode and Frequency Range

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×16	1/4	1/8	1/8	1/4	1/4	4	2	2	4	4	10	40	20	20	40	40
	1/4	1/4	1/8	1/4	1/4	4	4	2	4	4		40	40	20	40	40
	1/4	1/4	1/4	1/4	1/4	4	4	4	4	4		40	40	40	40	40
	1/2	1/8	1/8	1/4	1/4	8	2	2	4	4		80	20	20	40	40
	1/2	1/8	1/8	1/2	1/4	8	2	2	8	4		80	20	20	80	40
	1/2	1/4	1/8	1/4	1/4	8	4	2	4	4		80	40	20	40	40
	1/2	1/4	1/8	1/2	1/4	8	4	2	8	4		80	40	20	80	40
	1/2	1/4	1/4	1/4	1/4	8	4	4	4	4		80	40	40	40	40
	1/2	1/4	1/4	1/2	1/4	8	4	4	8	4		80	40	40	80	40
	1/1	1/8	1/8	1/4	1/4	16	2	2	4	4		160	20	20	40	40
	1/1	1/8	1/8	1/2	1/4	16	2	2	8	4		160	20	20	80	40
	1/1	1/4	1/8	1/4	1/4	16	4	2	4	4		160	40	20	40	40
	1/1	1/4	1/8	1/2	1/4	16	4	2	8	4		160	40	20	80	40
	1/1	1/4	1/4	1/4	1/4	16	4	4	4	4		160	40	40	40	40
	1/1	1/4	1/4	1/2	1/4	16	4	4	8	4		160	40	40	80	40

PLL Multiplication Ratio	FRQCR/MCLKCR/ACLKCR Division Ratio Setting					Clock Ratio					Clock Frequency (MHz)*					
	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ	Input Clock	Iϕ	Bϕ	Pϕ	Mϕ	Aϕ
×16	1/4	1/8	1/8	1/4	1/4	4	2	2	4	4	12.5	50	25	25	50	50
	1/4	1/4	1/8	1/4	1/4	4	4	2	4	4		50	50	25	50	50
	1/4	1/4	1/4	1/4	1/4	4	4	4	4	4		50	50	50	50	50
	1/2	1/8	1/8	1/4	1/4	8	2	2	4	4		100	25	25	50	50
	1/2	1/8	1/8	1/2	1/4	8	2	2	8	4		100	25	25	100	50
	1/2	1/4	1/8	1/4	1/4	8	4	2	4	4		100	50	25	50	50
	1/2	1/4	1/8	1/2	1/4	8	4	2	8	4		100	50	25	100	50
	1/2	1/4	1/4	1/4	1/4	8	4	4	4	4		100	50	50	50	50
	1/2	1/4	1/4	1/2	1/4	8	4	4	8	4		100	50	50	100	50
	1/1	1/8	1/8	1/4	1/4	16	2	2	4	4		200	25	25	50	50
	1/1	1/8	1/8	1/2	1/4	16	2	2	8	4		200	25	25	100	50
	1/1	1/4	1/8	1/4	1/4	16	4	2	4	4		200	50	25	50	50
	1/1	1/4	1/8	1/2	1/4	16	4	2	8	4		200	50	25	100	50
	1/1	1/4	1/4	1/4	1/4	16	4	4	4	4		200	50	50	50	50
	1/1	1/4	1/4	1/2	1/4	16	4	4	8	4		200	50	50	100	50

Notes: * Clock frequencies when the input clock frequency is assumed to be the shown value.

1. The PLL multiplication ratio is fixed at ×16. The division ratio can be selected from ×1, ×1/2, ×1/4, and ×1/8 for each clock by the setting in the frequency control register.
2. The output frequency of the PLL circuit is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin and the multiplication ratio (×16) of the PLL circuit. This output frequency must be 200 MHz or lower.
3. The input to the divider is always the output from the PLL circuit.
4. The internal clock (Iϕ) frequency is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×16) of the PLL circuit, and the division ratio of the divider. The resultant frequency of the internal clock (Iϕ) must not exceed 200 MHz or 100 MHz (maximum operating frequency) or lower.
5. The bus clock (Bϕ) frequency is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×16) of the PLL circuit, and the division ratio of the divider. The resultant frequency of the bus clock (Bϕ) must not exceed 50 MHz or the internal clock (Iϕ) frequency.
6. The peripheral clock (Pϕ) frequency is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×16) of the PLL circuit, and the division ratio of the divider. The resultant frequency of the peripheral clock (Pϕ) must not exceed 50 MHz or the bus clock (Bϕ) frequency.

7. When using the MTU2S, the MTU2S clock (Mϕ) frequency must not exceed 100 MHz and exceed the Pϕ and Bϕ frequencies. The MTU2S clock (Mϕ) frequency is obtained by multiplication of the frequency of the input from the crystal resonator or EXTAL pin, the multiplication ratio (×16) of the PLL circuit, and the division ratio of the divider.
8. The frequency of the CK pin output is always equal to the bus clock (Bϕ) frequency.
9. When using the AD, the AD clock (Aϕ) frequency must be equal to or higher than the peripheral clock (Pϕ) frequency.
10. When using the USB, the peripheral clock (Pϕ) frequency must be 13 MHz or higher.
11. Uϕ must be fixed to 48 MHz. When generating Uϕ from the divider, input the clock 12 MHz or connect the crystal resonator of 12MHz to the EXTAL or XTAL.

4.4 Register Descriptions

The clock pulse generator has the following registers.

Table 4.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0535	H'FFFE0010	16
MTU2S clock frequency control register	MCLKCR	R/W	H'43	H'FFFE0410	8
AD clock frequency control register	ACLKCR	R/W	H'43	H'FFFE0414	8
Oscillation stop detection control register	OSCCR	R/W	H'00	H'FFFE001C	8

4.4.1 Frequency Control Register (FRQCR)

FRQCR is a 16-bit readable/writable register used to specify the frequency division ratios for the internal clock (Iϕ), bus clock (Bϕ), and peripheral clock (Pϕ). FRQCR is only accessible in word units. After setting FRQCR to a new value, read it to confirm that it actually holds the new value, then execute NOP instructions for 32 cycles of Pϕ. Additionally, make settings for individual modules after setting FRQCR.

FRQCR is initialized to H'0535 only by a power-on reset. FRQCR retains its previous value by a manual reset or in software standby mode. The previous value is also retained when an internal reset is triggered by an overflow of the WDT.

When switching the division ratio of bus clock frequency, the CK pin is fixed at low level for a cycle of an input clock so as to prevent a hazard of switching. To change the frequency, see section 4.5, Changing the Frequency.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	STC[2:0]			-	IFC[2:0]			-	PFC[2:0]		
Initial value:	0	0	0	0	0	1	0	1	0	0	1	1	0	1	0	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	STC[2:0]	101	R/W	Bus Clock (Bφ) Frequency Division Ratio These bits specify the frequency division ratio of the bus clock. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 Others: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	IFC[2:0]	011	R/W	Internal Clock (Iφ) Frequency Division Ratio These bits specify the frequency division ratio of the internal clock. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 Others: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PFC[2:0]	101	R/W	Peripheral Clock (Pϕ) Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 Others: Setting prohibited

4.4.2 MTU2S Clock Frequency Control Register (MCLKCR)

MCLKCR is an 8-bit readable/writable register. MCLKCR can be accessed only in byte units.

MCLKCR is initialized to H'43 only by a power-on reset. MCLKCR retains its previous value by a manual reset or in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MSDIVS[1:0]	
Initial value:	0	1	0	0	0	0	1	1
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	MSDIVS[1:0]	11	R/W	Division Ratio Select These bits specify the frequency division ratio of the source clock. Set these bits so that the output clock is 100 MHz or less, and also an integer multiple of the peripheral clock frequency (Pϕ). 00: × 1 01: × 1/2 10: Setting prohibited 11: × 1/4

4.4.3 AD Clock Frequency Control Register (ACLKCR)

ACLKCR is an 8-bit readable/writable register that can be accessed only in byte units. ACLKCR is initialized to H'43 only by a power-on reset, but retains its previous value by a manual reset or in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	ASDIVS[1:0]	
Initial value:	0	1	0	0	0	0	1	1
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	ASDIVS[1:0]	11	R/W	Division Ratio Select These bits specify the frequency division ratio of the source clock. Set these bits so that the output clock is 50 MHz or less, and also an integer multiple of the peripheral clock frequency (Pϕ). 00: $\times$ 1 01: $\times$ 1/2 10: Setting prohibited 11: $\times$ 1/4

4.4.4 Oscillation Stop Detection Control Register (OSCCR)

OSCCR is an 8-bit readable/writable register that has an oscillation stop detection flag and selects flag status output to an external pin. OSCCR can be accessed only in byte units.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	OSC STOP	-	OSC ERS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	OSCSTOP	0	R/W	Oscillation Stop Detection Flag [Setting condition] • When a stop in the clock input is detected during normal operation [Clearing condition] • By a power-on reset input through the $\overline{\text{RES}}$ pin
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	OSCERS	0	R/W	Oscillation Stop Detection Flag Output Select Selects whether to output the oscillation stop detection flag signal through the $\overline{\text{WDTOVF}}$ pin. 0: Outputs only the WDT overflow signal through the $\overline{\text{WDTOVF}}$ pin 1: Outputs the WDT overflow signal and oscillation stop detection flag signal through the $\overline{\text{WDTOVF}}$ pin

4.5 Changing the Frequency

Selecting division ratios for the frequency divider can change the frequencies of the internal clock, bus clock, peripheral clock, MTU2S clock, and AD clock under the software control through the frequency control register (FRQCR), MTU2S clock frequency control register (MCKCR), and AD clock frequency control register (ACLKCR). The following describes how to specify the frequencies.

1. In the initial state, IFC2 to IFC0 = B'011 (×1/4), STC2 to STC0 = B'101 (×1/8), PFC2 to PFC0 = B'101 (×1/8), MSDIVS1 and MSDIVS0 = 11 (×1/4), and ASDIVS1 and ASDIVS 0 = 11 (×1/4).
2. Stop all modules except the CPU, on-chip ROM, and on-chip RAM.
3. Set the desired values in bits IFC2 to IFC0, STC2 to STC0, PFC2 to PFC0, MSDIVS1, MSDIVS0, ASDIVS1, and ASDIVS 0. When specifying the frequencies, satisfy the following condition: internal clock (Iϕ) ≥ bus clock (Bϕ) ≥ peripheral clock (Pϕ). When using the MTU2S clock, specify the frequencies to satisfy the following condition: 100 MHz ≥ MTU2S clock (MIϕ) ≥ peripheral clock (Pϕ).
4. The clock frequencies are immediately changed to the specified values after FRQCR setting is completed.
5. When changing the frequency division ratio for Bϕ after having set the ratios for Bϕ and Pϕ to 1/4 or a higher value, follow the procedure below rather than simultaneously changing the ratios for Iϕ, Bϕ, and Pϕ.
   1. Change only the ratio of Pϕ to 1/8 (PFC in FRQCR = B'101).
   2. After switching the setting for Pϕ, set only the ratio for Bϕ to the desired value.
   3. Set the ratios for Iϕ and Pϕ to the desired values.

   The limitation only applies to changes to the ratio for Bϕ. No limitation applies to procedures for changing Iϕ and Pϕ. Furthermore, no limitation applies to procedures for changing the ratios for Iϕ, Bϕ, and Pϕ from the initial values to desired values. Simultaneously changing settings for Iϕ, Bϕ, and Pϕ is possible. Note that FRQCR values should be changed by program code in the on-chip RAM. Even if FRQCR values are changed from initial ones. It is also changed by program code in the on-chip RAM.

4.6 Oscillator

The source of click supply can be selected from a connected crystal resonator or an external clock input through a pin.

4.6.1 Connecting Crystal Resonator

A crystal resonator can be connected as shown in figure 4.2. Use the damping resistance (R_d) shown in table 4.5. Use a crystal resonator that has a resonance frequency of 10 to 12.5 MHz.

It is recommended to consult the crystal resonator manufacturer concerning the compatibility of the crystal resonator and the LSI.

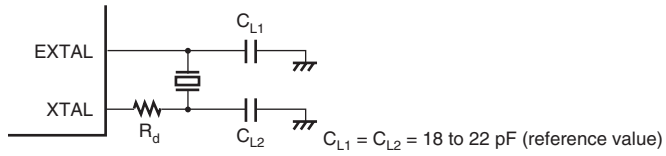


Figure 4.2 Example of Crystal Resonator Connection

Table 4.5 Damping Resistance Values (Reference Values)

Frequency (MHz)	10	12.5
R_d (Ω) (reference value)	0	0

Figure 4.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator with the characteristics shown in table 4.6.

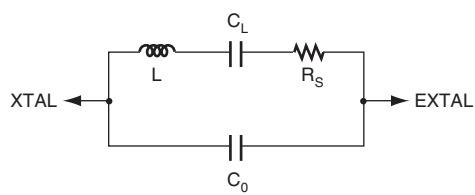


Figure 4.3 Crystal Resonator Equivalent Circuit

Table 4.6 Crystal Resonator Characteristics

Frequency (MHz)	10	12.5
R_s max. (Ω) (reference value)	60	50
C_0 max. (pF) (reference value)	7	7

4.6.2 External Clock Input Method

Figure 4.4 shows an example of an external clock input connection. Drive the external clock high when it is stopped in software standby mode. During operation, input an external clock with a frequency of 10 to 12.5 MHz. Make sure the parasitic capacitance of the XTAL pin is 10 pF or less.

Even when inputting an external clock, be sure to wait at least for the oscillation settling time in power-on sequence or in canceling software standby mode, in order to ensure the PLL settling time.



Figure 4.4 Example of External Clock Connection

4.7 Oscillation Stop Detection

The CPG detects a stop in the clock input if any system abnormality halts the clock supply.

When no change has been detected in the EXTAL input for a certain period, the OSCSTOP bit in OSCCR is set to 1 and this state is retained until a power-on reset is input through the $\overline{\text{RES}}$ pin is canceled. If the OSCERS bit is 1 at this time, an oscillation stop detection flag signal is output through the $\overline{\text{WDTOVF}}$ pin. In addition, the high-current ports (multiplexed pins to which the TIOC3B, TIOC3D, and TIOC4A to TIOC4D signals in the MTU2, the TIOC3BS, TIOC3DS, and TIOC4AS to TIOC4DS in the MTU2S are assigned) can be placed in high-impedance state regardless of settings of the OSCERS bit and PFC.

Even in software standby mode, these pins can be placed in high-impedance state. For details, refer to appendix A, Pin States. Under an abnormal condition where oscillation stops while the LSI is not in software standby mode, LSI operations other than the oscillation stop detection function become unpredictable. In this case, even after oscillation is restarted, LSI operations including the above high-current pins become unpredictable.

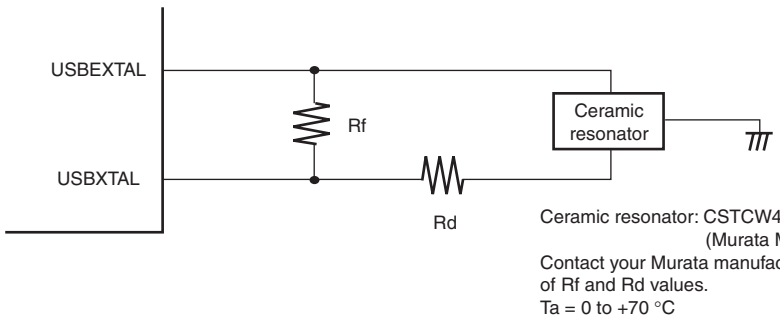
Even while no change is detected in the EXTAL input, the PLL circuit in this LSI continues oscillating at a frequency range from 100 kHz to 10 MHz (depending on the temperature and operating voltage).

4.8 USB Operating Clock (48 MHz)

Connection of a ceramic resonator for USB, input of an external 48-MHz clock signal, and selection of the internal CPG are available as methods for supplying the USB operating clock.

4.8.1 Connecting a Ceramic Resonator

Figure 4.5 shows an example of the connections for a ceramic resonator.



Note: *** represents a three-digit alphanumeric which express " Individual Specification". Since the frequency for USB requires high accuracy, the official product name will be decided to match the frequency after evaluation of oscillation on the board that is actually to be used.
Please contact your Renesas Electronics sales agency.

Figure 4.5 Example of Connecting a Ceramic Resonator

4.8.2 Input of an External 48-MHz Clock Signal

Figure 4.6 shows an example of the connections for input of an external 48-MHz clock signal. The USBXTAL pin must be left open.

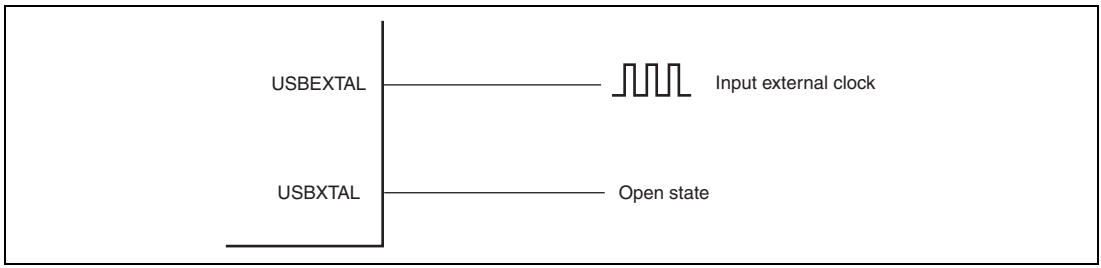


Figure 4.6 Example of Connecting an External 48-MHz Clock

Table 4.7 shows the input conditions for the external 48-MHz clock.

Table 4.7 Input Conditions for the External 48-MHz Clock

Item	Symbol	Min.	Max.	Unit	Reference Figure
Frequency (48 MHz)	t_{FREQ}	47.88	48.12	MHz	Figure 4.7
Clock rise time	t_{R48}	—	3	ns	
Clock fall time	t_{F48}	—	3	ns	
Duty (t_{HIGH}/t_{FREQ})	t_{DUTY}	40	60	%	

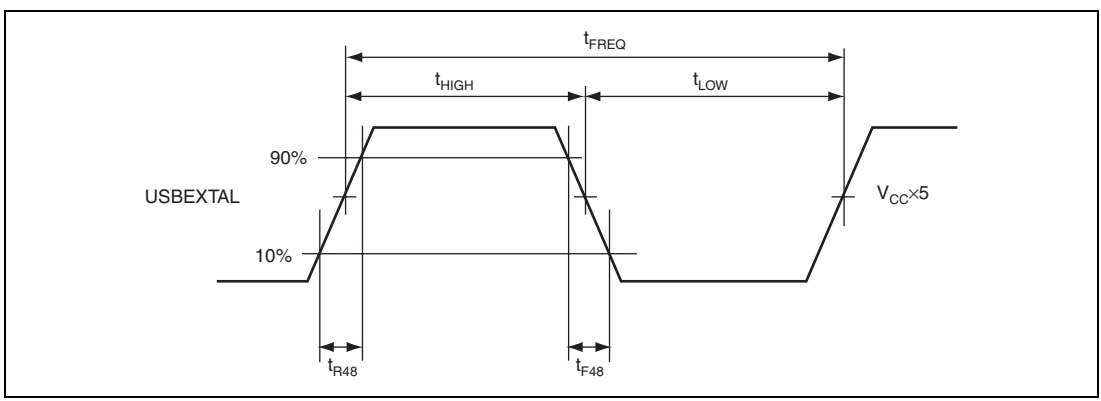


Figure 4.7 Input Timing of External 48-MHz Clock

4.8.3 Handling of pins when a Ceramic Resonator is not Connected (the Internal CPG is Selected or the USB is Not in Use)

When a ceramic resonator is not connected, connect the USBEXTAL pin to ground (Vss) and leave the USBEXTAL pin open-circuit as shown in figure 4.8. Possible clock frequencies for input to EXTAL are fixed to 12 MHz. We recommend a 4-layer circuit board.

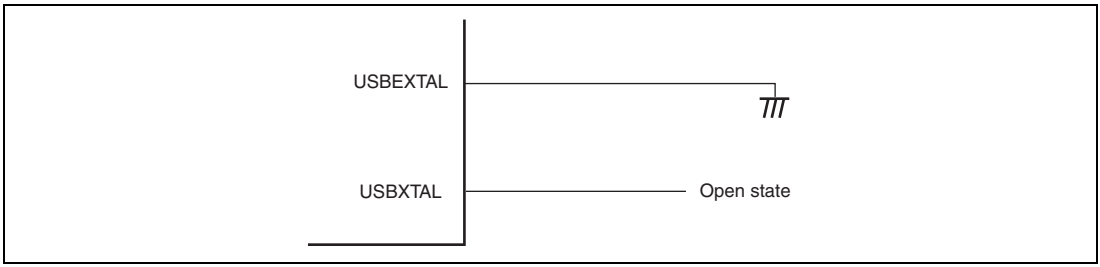
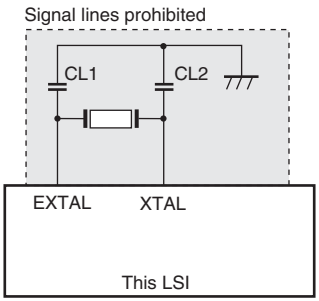


Figure 4.8 Handling of Pins when a Ceramic Resonator is not Connected

4.9 Notes on Board Design

4.9.1 Note on Using an External Crystal Resonator

Place the crystal resonator and capacitors CL1 and CL2 as close to the XTAL and EXTAL pins as possible. In addition, to minimize induction and thus obtain oscillation at the correct frequency, the capacitors to be attached to the resonator must be grounded to the same ground. Do not bring wiring patterns close to these components.

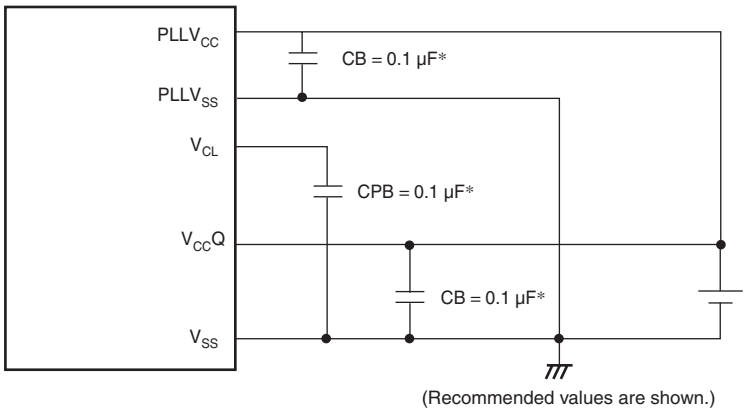


Reference value
CL1 = 20 pF
CL2 = 20 pF

Note: The values for CL1 and CL2 should be determined after consultation with the crystal resonator manufacturer.

Figure 4.9 Note on Using a Crystal Resonator

A circuitry shown in figure 4.10 is recommended as an external circuitry around the PLL. $PLLV_{CC}$, $PLLV_{SS}$, V_{CL}, and V_{SS} must be separated from the board power supply source to avoid an influence from power supply noise. Be sure to insert bypass capacitors CB and CPB close to the V_{CL} and V_{SS} pins. We recommend a 4-layer circuit board so that stable power-supply and ground levels are supplied to the LSI.



Note: * CB and CPB are laminated ceramic capacitors.

Figure 4.10 Recommended External Circuitry around PLL

Section 5 Exception Handling

5.1 Overview

5.1.1 Types of Exception Handling and Priority

Exception handling is started by sources, such as resets, address errors, register bank errors, interrupts, and instructions. Table 5.1 shows their priorities. When several exception handling sources occur at once, they are processed according to the priority shown.

Table 5.1 Types of Exception Handling and Priority Order

Type	Exception Handling		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error		
	DMAC address error		
Instruction	FPU exception		
	Integer division exception (division by zero)		
	Integer division exception (overflow)		
Register bank error	Bank underflow		
	Bank overflow		
Interrupt	NMI		
	User break		
	H-UDI		
	IRQ		
	Memory error (flash memory, data flash)		
	On-chip peripheral modules	A/D converter (ADC)	
		Controller area network (RCAN-ET)	
		Direct memory access controller (DMAC)	
		Compare match timer (CMT)	
		Bus state controller (BSC)	
		USB function module (USB) EP4 FIFO full/EP5 FIFO empty on DTC transfer end	Low

Type	Exception Handling		Priority
Interrupt	On-chip peripheral modules	Watchdog timer (WDT)	High
		Ethernet controller (Ether-C, E-DMAC)	
		USB function module (USB) EP1 FIFO full/EP2 FIFO empty on DTC transfer end	
		Multi-function timer pulse unit 2 (MTU2)	
		Port output enable 2 (POE2): OEI1 and OEI2 interrupts	
		Multi-function timer pulse unit 2S (MTU2S)	
		Port output enable 2 (POE2): OEI3 interrupt	
		USB function module (USB) USI0/USI1	
		I^2C bus interface 3 (IIC3)	
		Renesas serial peripheral interface (RSPI)	
		Serial communication interface (SCI)	
		Serial communication interface with FIFO (SCIF)	
Instruction	Trap instruction (TRAPA instruction)		
	General illegal instructions (undefined code)		
	Slot illegal instructions (undefined code placed directly after a delayed branch instruction*[1], instructions that rewrite the PC*[2], 32-bit instructions*[3], RESBANK instruction, DIVS instruction, and DIVU instruction)		Low

Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.

2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N.

3. 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, FMOV.S@disp12, FMOV.D@disp12, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.

5.1.2 Exception Handling Operations

The exception handling sources are detected and begin processing according to the timing shown in table 5.2.

Table 5.2 Timing of Exception Source Detection and Start of Exception Handling

Exception	Source	Timing of Source Detection and Start of Handling
Reset	Power-on reset	Starts when the $\overline{\text{RES}}$ pin changes from low to high, when the H-UDI reset negate command is set after the H-UDI reset assert command has been set, or when the WDT overflows.
	Manual reset	Starts when the $\overline{\text{MRES}}$ pin changes from low to high or when the WDT overflows.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Register bank error	Bank underflow	Starts upon attempted execution of a RESBANK instruction when saving has not been performed to register banks.
	Bank overflow	In the state where saving has been performed to all register bank areas, starts when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except immediately after a delayed branch instruction (delay slot).
	Slot illegal instructions	Starts from the decoding of undefined code placed immediately after a delayed branch instruction (delay slot), of instructions that rewrite the PC, of 32-bit instructions, of the RESBANK instruction, of the DIVS instruction, or of the DIVU instruction.
	Integer division instructions	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by −1.
	Floating point operation instructions	Starts when detecting invalid operation exception defined by IEEE standard 754, division-by-zero exception, overflow, underflow, or inexact exception. Also starts when qNAN or ±∞ is input to the source for a floating point operation instruction when the QIS bit in FPSCR is set.

When exception handling starts, the CPU operates as follows:

(1) Exception Handling Triggered by Reset

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception handling vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 5.1.3, Exception Handling Vector Table, for more information. The vector base register (VBR) is then initialized to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the interrupt controller (INTC) is also initialized to 0. The program begins running from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Register Bank Errors, Interrupts, and Instructions

SR and PC are saved to the stack indicated by R15. In the case of interrupt exception handling other than NMI or UBC with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved to the register banks. In the case of exception handling due to an address error, register bank error, NMI interrupt, UBC interrupt, or instruction, saving to a register bank is not performed. When saving is performed to all register banks, automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception will be generated. In the case of interrupt exception handling, the interrupt priority level is written to the I3 to I0 bits in SR. In the case of exception handling due to an address error or instruction, the I3 to I0 bits are not affected. The start address is then fetched from the exception handling vector table and the program begins running from that address.

5.1.3 Exception Handling Vector Table

Before exception handling begins running, the exception handling vector table must be set in memory. The exception handling vector table stores the start addresses of exception service routines. (The reset exception handling table holds the initial values of PC and SP.)

All exception sources are given different vector numbers and vector table address offsets, from which the vector table addresses are calculated. During exception handling, the start addresses of the exception service routines are fetched from the exception handling vector table, which is indicated by this vector table address.

Table 5.3 shows the vector numbers and vector table address offsets. Table 5.4 shows how vector table addresses are calculated.

Table 5.3 Exception Handling Vector Table

Exception Sources		Vector Numbers	Vector Table Address Offset
Power-on reset	PC	0	H'00000000 to H'00000003
	SP	1	H'00000004 to H'00000007
Manual reset	PC	2	H'00000008 to H'0000000B
	SP	3	H'0000000C to H'0000000F
General illegal instruction		4	H'00000010 to H'00000013
(Reserved by system)		5	H'00000014 to H'00000017
Slot illegal instruction		6	H'00000018 to H'0000001B
(Reserved by system)		7	H'0000001C to H'0000001F
		8	H'00000020 to H'00000023
CPU address error		9	H'00000024 to H'00000027
DMAC address error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
FPU exception		13	H'00000034 to H'00000037
H-UDI		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043

Exception Sources	Vector Numbers	Vector Table Address Offset
Integer division exception (division by zero)	17	H'00000044 to H'00000047
Integer division exception (overflow)	18	H'00000048 to H'0000004B
(Reserved by system)	19	H'0000004C to H'0000004F
	:	:
	31	H'0000007C to H'0000007F
Trap instruction (user vector)	32	H'00000080 to H'00000083
	:	:
	63	H'000000FC to H'000000FF
External interrupts (IRQ), on-chip peripheral module interrupts*	64	H'00000100 to H'00000103
	:	:
	511	H'000007FC to H'000007FF

Note: * The vector numbers and vector table address offsets for each external interrupt and on-chip peripheral module interrupt are given in table 6.4 in section 6, Interrupt Controller (INTC).

Table 5.4 Calculating Exception Handling Vector Table Addresses

Exception Source	Vector Table Address Calculation
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, register bank errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

Notes: 1. Vector table address offset: See table 5.3.
2. Vector number: See table 5.3.

5.2 Resets

5.2.1 Types of Reset

A reset is the highest-priority exception handling source. There are two kinds of reset, power-on and manual. As shown in table 5.5, the CPU state is initialized in both a power-on reset and a manual reset. On-chip peripheral module registers are initialized by a power-on reset, but not by a manual reset.

Table 5.5 Exception Source Detection and Exception Handling Start Timing

	Conditions for Transition to Reset State			Internal States		
Type	$\overline{\text{RES}}$ or $\overline{\text{MRES}}$	H-UDI Command	WDT Overflow	CPU, FPU	On-Chip Peripheral Modules, I/O Port	WRCSR of WDT, FRQCR of CPG
Power-on reset	Low	—	—	Initialized	Initialized	Initialized
	High	H-UDI reset assert command is set	—	Initialized	Initialized	Initialized
	High	Command other than H-UDI reset assert is set	Power-on reset	Initialized	Initialized	Not initialized
Manual reset	Low	—	—	Initialized	Not initialized*	Not initialized
	High	—	Manual reset	Initialized	Not initialized*	Not initialized

Note: * The BN bit in IBNR of the INTC is initialized.

5.2.2 Power-On Reset

(1) Power-On Reset by Means of $\overline{\text{RES}}$ Pin

When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept at the low level for the duration of the oscillation settling time at power-on or when in software standby mode (when the clock is halted), or at least 20 t_{cyc} when the clock is running. In the power-on reset state, the internal state of the CPU and all the on-chip peripheral module registers are initialized. See appendix A, Pin States, for the status of individual pins during the power-on reset state.

In the power-on reset state, power-on reset exception handling starts when the $\overline{\text{RES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

Be certain to always perform power-on reset processing when turning the system power on.

(2) Power-On Reset by Means of H-UDI Reset Assert Command

When the H-UDI reset assert command is set, this LSI enters the power-on reset state. Power-on reset by means of an H-UDI reset assert command is equivalent to power-on reset by means of the $\overline{\text{RES}}$ pin. Setting the H-UDI reset negate command cancels the power-on reset state. The time required between an H-UDI reset assert command and H-UDI reset negate command is the same as the time to keep the $\overline{\text{RES}}$ pin low to initiate a power-on reset. In the power-on reset state generated by an H-UDI reset assert command, setting the H-UDI reset negate command starts power-on reset exception handling. The CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

(3) Power-On Reset Initiated by WDT

When a setting is made for a power-on reset to be generated in the WDT's watchdog timer mode, and WTCNT of the WDT overflows, this LSI enters the power-on reset state.

In this case, WRCSR of the WDT and FRQCR of the CPG are not initialized by the reset signal generated by the WDT.

If a reset caused by the $\overline{\text{RES}}$ pin or the H-UDI reset assert command occurs simultaneously with a reset caused by WDT overflow, the reset caused by the $\overline{\text{RES}}$ pin or the H-UDI reset assert command has priority, and the WOVF bit in WRCSR is cleared to 0. When power-on reset exception processing is started by the WDT, the CPU operates in the same way as when a power-on reset was caused by the $\overline{\text{RES}}$ pin.

5.2.3 Manual Reset

(1) Manual Reset by Means of $\overline{\text{MRES}}$ Pin

When the $\overline{\text{MRES}}$ pin is driven low, this LSI enters the manual reset state. To reset this LSI without fail, the $\overline{\text{MRES}}$ pin should be kept at the low level for at least 20 t_{cyc}. In the manual reset state, the CPU's internal state is initialized, but all the on-chip peripheral module registers are not initialized. In the manual reset state, manual reset exception handling starts when the $\overline{\text{MRES}}$ pin is first driven low for a fixed period and then returned to high. The CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception handling vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception handling vector table.
3. The vector base register (VBR) is cleared to H'00000000, the interrupt mask level bits (I3 to I0) of the status register (SR) are initialized to H'F (B'1111), and the BO and CS bits are initialized. The BN bit in IBNR of the INTC is also initialized to 0.
4. The values fetched from the exception handling vector table are set in the PC and SP, and the program begins executing.

(2) Manual Reset Initiated by WDT

When a setting is made for a manual reset to be generated in the WDT's watchdog timer mode, and WTCNT of the WDT overflows, this LSI enters the manual reset state.

When manual reset exception processing is started by the WDT, the CPU operates in the same way as when a manual reset was caused by the $\overline{\text{MRES}}$ pin.

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception handling will be deferred until the CPU acquires the bus.

5.3 Address Errors

5.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 5.6.

Table 5.6 Bus Cycles and Address Errors

Bus Cycle			
Type	**Bus Master**	**Bus Cycle Description**	**Address Errors**
Instruction fetch	CPU	Instruction fetched from even address	None (normal)
		Instruction fetched from odd address	Address error occurs
		Instruction fetched from other than on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	None (normal)
		Instruction fetched from on-chip peripheral module space* or H'F0000000 to H'F5FFFFFF in on-chip RAM space*	Address error occurs
		Instruction fetched from external memory space in single-chip mode	Address error occurs
Data read/write	CPU, DMAC, DTC or E-DMAC	Word data accessed from even address	None (normal)
		Word data accessed from odd address	Address error occurs
		Longword data accessed from a longword boundary	None (normal)
		Longword data accessed from other than a long-word boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space*	None (normal)
		Double longword data accessed from a double longword boundary	None (normal)
		Double Longword data accessed from other than a double longword boundary	Address error occurs

Bus Cycle			
Type	**Bus Master**	**Bus Cycle Description**	**Address Errors**
Data read/write	CPU, DMAC, DTC or E-DMAC	Longword data accessed in 16-bit on-chip peripheral module space*	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*	None (normal)
		External memory space accessed when in single chip mode	Address error occurs

Note: * See section 9, Bus State Controller (BSC), for details of the on-chip peripheral module space and on-chip RAM space.

5.3.2 Address Error Exception Handling

When an address error occurs, the bus cycle in which the address error occurred ends*. When the executing instruction then finishes, address error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the address error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

Note: * In the case of an address error caused by instruction fetching when data is read or written, if the bus cycle on which the address error occurred is not completed by the end of the operations described above operation 3, the CPU will recommence address error exception processing until the end of that bus cycle.

5.4 Register Bank Errors

5.4.1 Register Bank Error Sources

(1) Bank Overflow

In the state where saving has already been performed to all register bank areas, bank overflow occurs when acceptance of register bank overflow exception has been set by the interrupt controller (the BOVE bit in IBNR of the INTC is set to 1) and an interrupt that uses a register bank has occurred and been accepted by the CPU.

(2) Bank Underflow

Bank underflow occurs when an attempt is made to execute a RESBANK instruction while saving has not been performed to register banks.

5.4.2 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a bank overflow, and the start address of the executed RESBANK instruction for a bank underflow.

   To prevent multiple interrupts from occurring at a bank overflow, the interrupt priority level that caused the bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.5 Interrupts

5.5.1 Interrupt Sources

Table 5.7 shows the sources that start up interrupt exception handling. These are divided into NMI, user breaks, H-UDI, IRQ, memory errors, and on-chip peripheral modules.

Table 5.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	User debugging interface (H-UDI)	1
IRQ	IRQ0 to IRQ7 pins (external input)	8
Memory error	Flash memory (ROM), data flash (FLD)	1
On-chip peripheral module	A/D converter (ADC)	2
	Controller area network (RCAN-ET)	4
	Direct memory access controller (DMAC)	16
	Compare match timer (CMT)	2
	Bus state controller (BSC)	1
	Watchdog timer (WDT)	1
	Ethernet controller (Ether-C, E-DMAC)	1
	USB function module (USB)	6
	Multi-function timer pulse unit 2 (MTU2)	28
	Multi-function timer pulse unit 2S (MTU2S)	13
	Port output enable 2 (POE2)	3
	I^2C bus interface 3 (IIC3)	5
	Renesas serial peripheral interface (RSPI)	3
	Serial communication interface (SCI)	16
	Serial communication interface with FIFO (SCIF)	4

Each interrupt source is allocated a different vector number and vector table offset. See table 6.4 in section 6, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

5.5.2 Interrupt Priority Level

The interrupt priority order is predetermined. When multiple interrupts occur simultaneously (overlap), the interrupt controller (INTC) determines their relative priorities and starts processing according to the results.

The priority order of interrupts is expressed as priority levels 0 to 16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break interrupt and H-UDI interrupt priority level is 15. Priority levels of IRQ interrupts, and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01, 02, and 05 to 19 (IPR01, IPR02, and IPR05 to IPR19) of the INTC as shown in table 5.8. The priority levels that can be set are 0 to 15. Level 16 cannot be set. See section 6.3.1, Interrupt Priority Registers 01, 02, 05 to 19 (IPR01, IPR02, IPR05 to IPR19), for details of IPR01, IPR02, and IPR05 to IPR19.

Table 5.8 Interrupt Priority Order

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
H-UDI	15	Fixed priority level.
IRQ	0 to 15	Set with interrupt priority registers (IPR).
On-chip peripheral module		
Memory error	15	Fixed priority level.

5.5.3 Interrupt Exception Handling

When an interrupt occurs, its priority level is ascertained by the interrupt controller (INTC). NMI is always accepted, but other interrupts are only accepted if they have a priority level higher than the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR).

When an interrupt is accepted, interrupt exception handling begins. In interrupt exception handling, the CPU fetches the exception service routine start address which corresponds to the accepted interrupt from the exception handling vector table, and saves SR and the program counter (PC) to the stack. In the case of interrupt exception handling other than NMI or UBC with usage of the register banks enabled, general registers R0 to R14, control register GBR, system registers MACH, MACL, and PR, and the vector table address offset of the interrupt exception handling to be executed are saved in the register banks. In the case of exception handling due to an address error, NMI interrupt, UBC interrupt, or instruction, saving is not performed to the register banks. If saving has been performed to all register banks (0 to 14), automatic saving to the stack is performed instead of register bank saving. In this case, an interrupt controller setting must have been made so that register bank overflow exceptions are not accepted (the BOVE bit in IBNR of the INTC is 0). If a setting to accept register bank overflow exceptions has been made (the BOVE bit in IBNR of the INTC is 1), register bank overflow exception occurs. Next, the priority level value of the accepted interrupt is written to the I3 to I0 bits in SR. For NMI, however, the priority level is 16, but the value set in the I3 to I0 bits is H'F (level 15). Then, after jumping to the start address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch. See section 6.6, Operation, for further details of interrupt exception handling.

5.6 Exceptions Triggered by Instructions

5.6.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, slot illegal instructions, general illegal instructions, integer division exceptions, and floating-point operation instructions, as shown in table 5.9.

Table 5.9 Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instruction	TRAPA	
Slot illegal instructions	Undefined code placed immediately after a delayed branch instruction (delay slot), instructions that rewrite the PC, 32-bit instructions, RESBANK instruction, DIVS instruction, and DIVU instruction	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF, JSR/N, RTV/N 32-bit instructions: BAND.B, BANDNOT.B, BCLR.B, BLD.B, BLDNOT.B, BOR.B, BORNOT.B, BSET.B, BST.B, BXOR.B, MOV.B@disp12, MOV.W@disp12, FMOV.S@disp12, FMOV.D@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.
General illegal instructions	Undefined code anywhere besides in a delay slot	
Integer division exceptions	Division by zero	DIVU, DIVS
	Negative maximum value ÷ (−1)	DIVS
Floating-point operation instructions	Starts when detecting invalid operation exception defined by IEEE754, division-by-zero exception, overflow, underflow, or inexact exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

5.6.2 Trap Instructions

When a TRAPA instruction is executed, trap instruction exception handling starts. The CPU operates as follows:

1. The exception service routine start address which corresponds to the vector number specified in the TRAPA instruction is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.6.3 Slot Illegal Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code, an instruction that rewrites the PC, a 32-bit instruction, an RESBANK instruction, a DIVS instruction, or a DIVU instruction, slot illegal exception handling starts when such kind of instruction is decoded. The CPU operates as follows:

1. The exception service routine start address is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code, the instruction that rewrites the PC, the 32-bit instruction, the RESBANK instruction, the DIVS instruction, or the DIVU instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.6.4 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch instruction (i.e., in a delay slot) is decoded, general illegal instruction exception handling starts. The CPU handles general illegal instructions in the same way as slot illegal instructions. Unlike processing of slot illegal instructions, however, the program counter value stored is the start address of the undefined code.

5.6.5 Integer Division Instructions

When an integer division instruction performs division by zero or the result of integer division overflows, integer division instruction exception handling starts. The instructions that may become the source of division-by-zero exception are DIVU and DIVS. The only source instruction of overflow exception is DIVS, and overflow exception occurs only when the negative maximum value is divided by −1. The CPU operates as follows:

1. The exception service routine start address which corresponds to the integer division instruction exception that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the integer division instruction at which the exception occurred.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

5.6.6 Floating Point Operation Instruction

An FPU exception is generated when the V, Z, O, U or I bit in the FPU enable field (Enable) of the floating point status/control register (FPSCR) is set. This indicates the occurrence of an invalid operation exception defined by the IEEE standard 754, a division-by-zero exception, overflow (in the case of an instruction for which this is possible), underflow (in the case of an instruction for which this is possible), or inexact exception (in the case of an instruction for which this is possible).

The instructions that may cause FPU exception are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

An FPU exception is generated only when the corresponding enable bit (Enable) is set. When the FPU detects an exception source, FPU operation is suspended and generation of the exception is reported to the CPU. When exception handling is started, the CPU operations are as follows.

1. The start address of the exception service routine corresponding to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not an FPU exception has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time a floating-point operation instruction is executed.

When the V bit in the FPU exception enable field (Enable) of FPSCR is set and the QIS bit in FPSCR is also set, an FPU exception is generated when qNaN or ±∞ is input to a floating point operation instruction source.

5.7 When Exception Sources Are Not Accepted

When an address error, register bank error (overflow), or interrupt is generated immediately after a delayed branch instruction, it is sometimes not accepted immediately but stored instead, as shown in table 5.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 5.10 Exception Source Generation Immediately after Delayed Branch Instruction

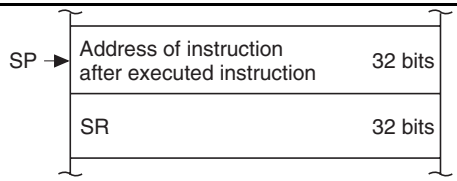
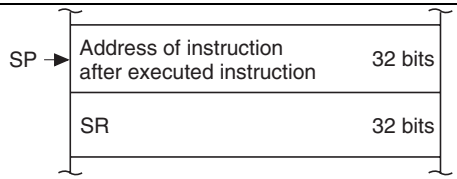
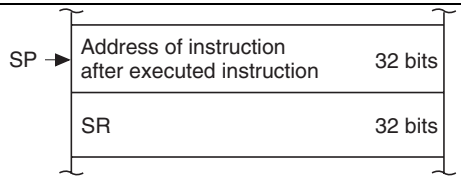
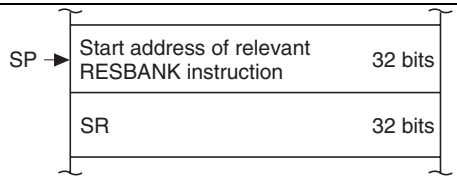
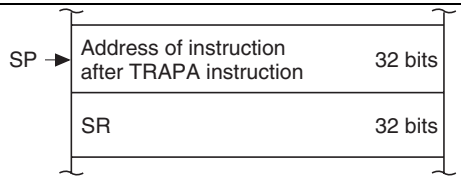
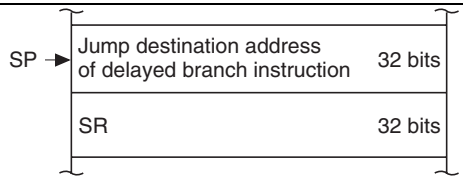
	Exception Source			
Point of Occurrence	**Address Error**	**FPU Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted

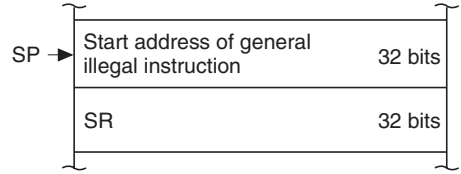
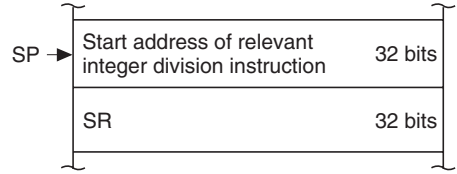
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

5.8 Stack Status after Exception Handling Ends

The status of the stack after exception handling ends is as shown in table 5.11.

Table 5.11 Stack Status After Exception Handling Ends

Exception Type	Stack Status
Address error	
Interrupt	
Register bank error (overflow) FPU exception	
Register bank error (underflow)	
Trap instruction	
Slot illegal instruction	

Exception Type	Stack Status
General illegal instruction	
Integer division instruction (division by zero, overflow)	

5.9 Usage Notes

5.9.1 Value of Stack Pointer (SP)

The value of the stack pointer must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

5.9.2 Value of Vector Base Register (VBR)

The value of the vector base register must always be a multiple of four. If it is not, an address error will occur when the stack is accessed during exception handling.

5.9.3 Address Errors Caused by Stacking of Address Error Exception Handling

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception handling (interrupts, etc.) and address error exception handling will start up as soon as the first exception handling is ended. Address errors will then also occur in the stacking for this address error exception handling. To ensure that address error exception handling does not go into an endless loop, no address errors are accepted at that point. This allows program control to be shifted to the address error exception service routine and enables error processing.

When an address error occurs during exception handling stacking, the stacking bus cycle (write) is executed. During stacking of the status register (SR) and program counter (PC), the SP is decremented by 4 for both, so the value of SP will not be a multiple of four after the stacking either. The address value output during stacking is the SP value, so the address where the error occurred is itself output. This means the write data stacked will be undefined.

5.9.4 Note When Changing Interrupt Mask Level (IMASK) of Status Register (SR) in CPU

When enabling or disabling interrupts by modifying the interrupt mask level value of the CPU status register (SR) using an LDC or LDC.L instruction, there must be at least five instructions between the instruction to enable interrupts and the instruction to disable interrupts.

Section 6 Interrupt Controller (INTC)

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

6.1 Features

- 16 levels of interrupt priority can be set

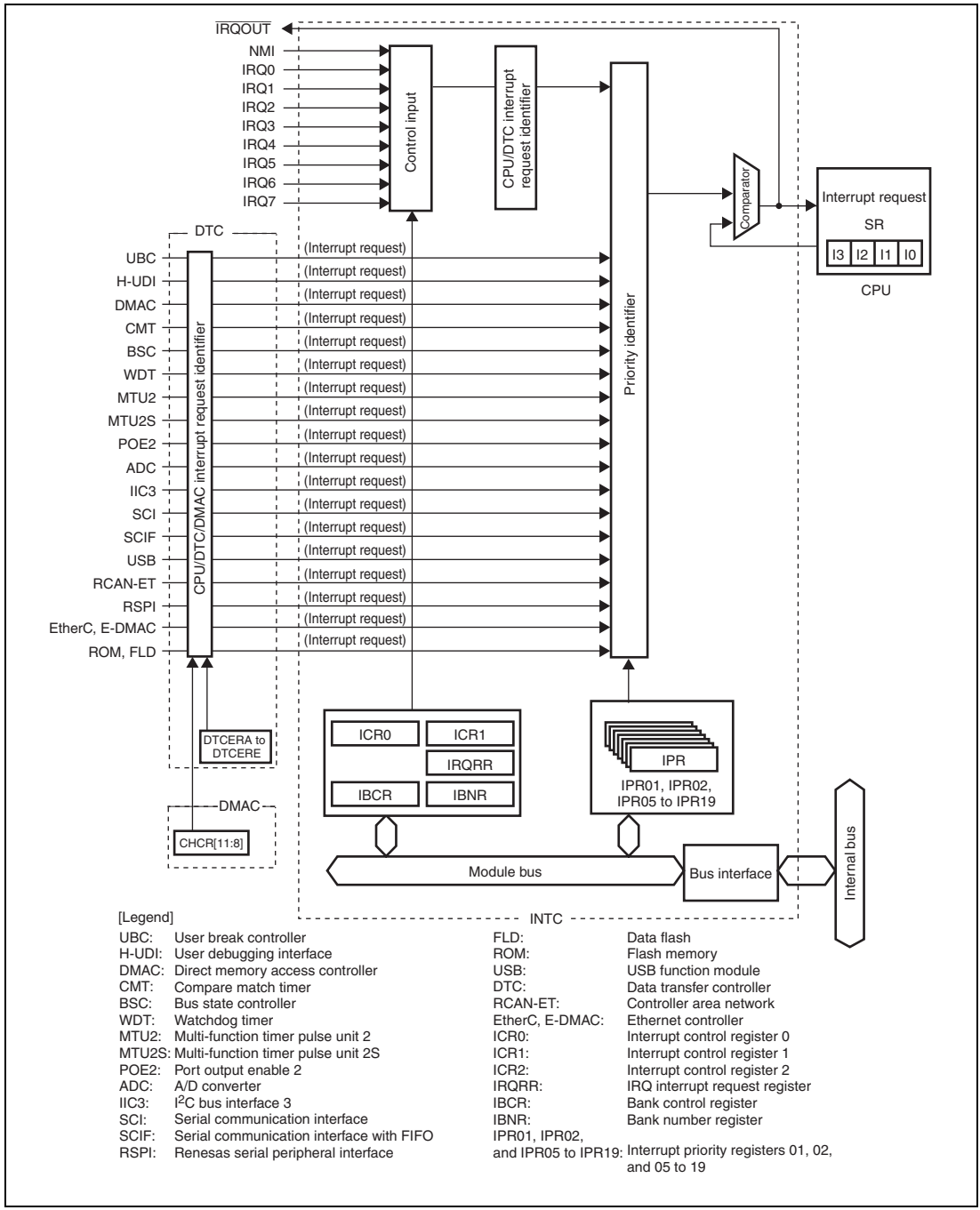
  By setting the 17 interrupt priority registers, the priority of IRQ interrupts and on-chip peripheral module interrupts can be selected from 16 levels for request sources.
- NMI noise canceler function

  An NMI input-level bit indicates the NMI pin state. By reading this bit in the interrupt exception service routine, the pin state can be checked, enabling it to be used as the noise canceler function.
- Occurrence of interrupt can be reported externally ($\overline{\text{IRQOUT}}$ pin)

  For example, when this LSI has released the bus mastership, this LSI can inform the external bus master of occurrence of an on-chip peripheral module interrupt and request for the bus mastership.
- Register banks

  This LSI has register banks that enable register saving and restoration required in the interrupt processing to be performed at high speed.

Figure 6.1 shows a block diagram of the INTC.



[Legend]

UBC:	User break controller	FLD:	Data flash
H-UDI:	User debugging interface	ROM:	Flash memory
DMAC:	Direct memory access controller	USB:	USB function module
CMT:	Compare match timer	DTC:	Data transfer controller
BSC:	Bus state controller	RCAN-ET:	Controller area network
WDT:	Watchdog timer	EtherC, E-DMAC:	Ethernet controller
MTU2:	Multi-function timer pulse unit 2	ICR0:	Interrupt control register 0
MTU2S:	Multi-function timer pulse unit 2S	ICR1:	Interrupt control register 1
POE2:	Port output enable 2	ICR2:	Interrupt control register 2
ADC:	A/D converter	IRQRR:	IRQ interrupt request register
IIC3:	I^2C bus interface 3	IBCR:	Bank control register
SCI:	Serial communication interface	IBNR:	Bank number register
SCIF:	Serial communication interface with FIFO	IPR01, IPR02, and IPR05 to IPR19:	Interrupt priority registers 01, 02, and 05 to 19
RSPI:	Renesas serial peripheral interface		

Figure 6.1 Block Diagram of INTC

6.2 Input/Output Pins

Table 6.1 shows the pin configuration of the INTC.

Table 6.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
Interrupt request input pins	IRQ7 to IRQ0	Input	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	Output	Output of signal to report occurrence of interrupt source

6.3 Register Descriptions

The INTC has the following registers. These registers are used to set the interrupt priorities and control detection of the external interrupt input signal.

Table 6.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*1	H'FFFE0800	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFE0802	16
IRQ interrupt request register	IRQRR	R/(W)*2	H'0000	H'FFFE0806	16
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	16
Interrupt priority register 01	IPR01	R/W	H'0000	H'FFFE0818	16, 32
Interrupt priority register 02	IPR02	R/W	H'0000	H'FFFE081A	16
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFE0820	16
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFE0C00	16, 32
Interrupt priority register 07	IPR07	R/W	H'0000	H'FFFE0C02	16
Interrupt priority register 08	IPR08	R/W	H'0000	H'FFFE0C04	16, 32
Interrupt priority register 09	IPR09	R/W	H'0000	H'FFFE0C06	16
Interrupt priority register 10	IPR10	R/W	H'0000	H'FFFE0C08	16, 32
Interrupt priority register 11	IPR11	R/W	H'0000	H'FFFE0C0A	16
Interrupt priority register 12	IPR12	R/W	H'0000	H'FFFE0C0C	16, 32
Interrupt priority register 13	IPR13	R/W	H'0000	H'FFFE0C0E	16
Interrupt priority register 14	IPR14	R/W	H'0000	H'FFFE0C10	16, 32
Interrupt priority register 15	IPR15	R/W	H'0000	H'FFFE0C12	16
Interrupt priority register 16	IPR16	R/W	H'0000	H'FFFE0C14	16, 32
Interrupt priority register 17	IPR17	R/W	H'0000	H'FFFE0C16	16
Interrupt priority register 18	IPR18	R/W	H'0000	H'FFFE0C18	16, 32
Interrupt priority register 19	IPR19	R/W	H'0000	H'FFFE0C1A	16
USB-DTC transfer interrupt request register	USDTENDRR	R/(W)*2	H'0000	H'FFFE0C50	16

Notes: Two access cycles are needed for word access, and four access cycles for longword access.

1. When the NMI pin is high, becomes H'8000; when low, becomes H'0000.
2. Only 0 can be written after reading 1, to clear the flag.

6.3.1 Interrupt Priority Registers 01, 02, 05 to 19 (IPR01, IPR02, IPR05 to IPR19)

IPR01, IPR02, and IPR05 to IPR19 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts and on-chip peripheral module interrupts. Table 6.3 shows the correspondence between the interrupt request sources and the bits in IPR01, IPR02, and IPR05 to IPR19.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 6.3 Interrupt Request Sources and IPR01, IPR02, and IPR05 to IPR19

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 01	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register 02	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register 05	Reserved	Reserved	ADI0	ADI1
Interrupt priority register 06	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 07	DMAC4	DMAC5	DMAC6	DMAC7
Interrupt priority register 08	CMT0	CMT1	BSC	WDT
Interrupt priority register 09	MTU2_0 (TGIA_0 to TGID_0)	MTU2_0 (TCIV_0, TGIE_0, TGIF_0)	MTU2_1 (TGIA_1, TGIB_1)	MTU2_1 (TCIV_1, TCIU_1)
Interrupt priority register 10	MTU2_2 (TGIA_2, TGIB_2)	MTU2_2 (TCIV_2, TCIU_2)	MTU2_3 (TGIA_3 to TGID_3)	MTU2_3 (TCIV_3)
Interrupt priority register 11	MTU2_4 (TGIA_4 to TGID_4)	MTU2_4 (TCIV_4)	MTU2_5 (TGIU_5, TGIV_5, TGIW_5)	POE2 (OEI1, OEI2)
Interrupt priority register 12	MTU2S_3 (TGIA_3S to TGID_3S)	MTU2S_3 (TCIV_3S)	MTU2S_4 (TGIA_4S to TGID_4S)	MTU2S_4 (TCIV_4S)

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 13	MTU2S_5 (TGIU_5S, TGIV_5S, TGIW_5S)	POE2 (OEI3)	IIC3	Reserved
Interrupt priority register 14	Reserved	Reserved	Reserved	SCIF3
Interrupt priority register 15	Reserved	Reserved	Reserved	Reserved
Interrupt priority register 16	SCI0	SCI1	SCI2	Reserved
Interrupt priority register 17	RSPI	SCI4	Reserved	Reserved
Interrupt priority register 18	USB (USI0, USI1)	RCAN-ET	EP1-FIFO full DTC transfer end (USBRXI0)	EP2-FIFO empty DTC transfer end (USBTXI0)
Interrupt priority register 19	EP4-FIFO full DTC transfer end (USBRXI1)	EP5-FIFO empty DTC transfer end (USBTXI1)	EtherC, E-DMAC (EINT0)	Reserved

As shown in table 6.3, by setting the 4-bit groups (bits 15 to 12, bits 11 to 8, bits 7 to 4, and bits 3 to 0) with values from H'0 (0000) to H'F (1111), the priority of each corresponding interrupt is set. Setting of H'0 means priority level 0 (the lowest level) and H'F means priority level 15 (the highest level).

IPR01, IPR02, and IPR05 to IPR19 are initialized to H'0000 by a power-on reset.

6.3.2 Interrupt Control Register 0 (ICR0)

ICR0 is a 16-bit register that sets the input signal detection mode for the external interrupt input pin NMI, and indicates the input level at the NMI pin.

ICR0 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	-	-	-	-	-	-	NMIE	-	-	-	-	-	-	-	-
Initial value:	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Note: * 1 when the NMI pin is high, and 0 when the NMI pin is low.

Bit	Bit Name	Initial Value	R/W	Description
15	NMIL	*	R	NMI Input Level Sets the level of the signal input at the NMI pin. The NMI pin level can be obtained by reading this bit. This bit cannot be modified. 0: Low level is input to NMI pin 1: High level is input to NMI pin
14 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	NMIE	0	R/W	NMI Edge Select Selects whether the falling or rising edge of the interrupt request signal on the NMI pin is detected. 0: Interrupt request is detected on falling edge of NMI input 1: Interrupt request is detected on rising edge of NMI input
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.3.3 Interrupt Control Register 1 (ICR1)

ICR1 is a 16-bit register that specifies the detection mode for external interrupt input pins IRQ7 to IRQ0 individually: low level, falling edge, rising edge, or both edges.

ICR1 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select
14	IRQ70S	0	R/W	These bits select whether interrupt signals corresponding to pins IRQ7 to IRQ0 are detected by a low level, falling edge, rising edge, or both edges.
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	00: Interrupt request is detected on low level of IRQn input
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	01: Interrupt request is detected on falling edge of IRQn input
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	10: Interrupt request is detected on rising edge of IRQn input
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	11: Interrupt request is detected on both edges of IRQn input
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

[Legend]

n = 7 to 0

Note: When the detecting condition of the IRQn input is changed, the IRQnF flag in IRQRR is cleared to 0.

6.3.4 IRQ Interrupt Request Register (IRQRR)

IRQRR is a 16-bit register that indicates interrupt requests from external input pins IRQ7 to IRQ0. If edge detection is set for the IRQ7 to IRQ0 interrupts, writing 0 to the IRQ7F to IRQ0F bits after reading IRQ7F to IRQ0F = 1 cancels the retained interrupts.

IRQRR is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	IRQ7F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ7 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred [Clearing condition] • IRQn input is high
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	

Bit	Bit Name	Initial Value	R/W	Description
0	IRQ0F	0	R/(W)*	1: IRQn interrupt has occurred [Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected [Clearing conditions] • Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF • Cleared by executing IRQn interrupt exception handling • Cleared when DTC is activated by the IRQn interrupt, then the DISEL bit in MRB of DTC is set to 0 • Cleared when the setting of IRQn1S or IRQn0S of ICR1 is changed 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin

[Legend]
n = 7 to 0

6.3.5 Bank Control Register (IBCR)

IBCR is a 16-bit register that enables or disables use of register banks for each interrupt priority level.

IBCR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15	E15	0	R/W	Enable These bits enable or disable use of register banks for interrupt priority levels 15 to 1. However, use of register banks is always disabled for the user break interrupts. 0: Use of register banks is disabled 1: Use of register banks is enabled
14	E14	0	R/W	
13	E13	0	R/W	
12	E12	0	R/W	
11	E11	0	R/W	
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

6.3.6 Bank Number Register (IBNR)

IBNR is a 16-bit register that enables or disables use of register banks and register bank overflow exception. IBNR also indicates the bank number to which saving is performed next through the bits BN3 to BN0.

IBNR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BE[1:0]		BOVE	-	-	-	-	-	-	-	-	-	BN[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BE[1:0]	00	R/W	Register Bank Enable These bits enable or disable use of register banks. 00: Use of register banks is disabled for all interrupts. The setting of IBCR is ignored. 01: Use of register banks is enabled for all interrupts except NMI and user break. The setting of IBCR is ignored. 10: Reserved (setting prohibited) 11: Use of register banks is controlled by the setting of IBCR.
13	BOVE	0	R/W	Register Bank Overflow Enable Enables of disables register bank overflow exception. 0: Generation of register bank overflow exception is disabled 1: Generation of register bank overflow exception is enabled
12 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	BN[3:0]	0000	R	Bank Number These bits indicate the bank number to which saving is performed next. When an interrupt using register banks is accepted, saving is performed to the register bank indicated by these bits, and BN is incremented by 1. After BN is decremented by 1 due to execution of a RESBANK (restore from register bank) instruction, restoration from the register bank is performed.

6.3.7 USB-DTC Transfer Interrupt Request Register (USDTENDRR)

USDTENDRR is a 16-bit register that indicates USB-DTC transfer end interrupt requests, which are on-chip peripheral module interrupts. Writing 0 to the RXF or TXF bit after reading RXF = 1 or TXF = 1 cancels the retained interrupt.

USDTENDRR is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXF0	TXF0	RXF1	TXF1	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R	R	R	R	R

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15	RXF0	0	R/(W)*	EP1-FIFO Full DTC Transfer End Interrupt Request (USBRXI0) 0: EP1-FIFO full DTC transfer end interrupt request has not occurred [Clearing conditions] • Cleared by reading RFX0 = 1, then writing 0 to RFX0 • Cleared by executing EP1-FIFO full DTC transfer end interrupt exception handling 1: EP1-FIFO full DTC transfer end interrupt request has occurred

Bit	Bit Name	Initial Value	R/W	Description
14	TXF0	0	R/(W)*	EP2-FIFO Empty DTC Transfer End Interrupt Request (USBTXI0) 0: EP2-FIFO empty DTC transfer end interrupt request has not occurred [Clearing conditions] • Cleared by reading TFX0 = 1, then writing 0 to TFX0 • Cleared by executing EP2-FIFO empty DTC transfer end interrupt exception handling 1: EP2-FIFO empty DTC transfer end interrupt request has occurred
13	RXF1	0	R/(W)*	EP4-FIFO Full DTC Transfer End Interrupt Request (USBRXI1) 0: EP4-FIFO full DTC transfer end interrupt request has not occurred [Clearing conditions] • Cleared by reading RFX1 = 1, then writing 0 to RFX1 • Cleared by executing EP4-FIFO full DTC transfer end interrupt exception handling 1: EP4-FIFO full DTC transfer end interrupt request has occurred
12	TXF1	0	R/(W)*	EP5-FIFO Empty DTC Transfer End Interrupt Request (USBTXI1) 0: EP5-FIFO empty DTC transfer end interrupt request has not occurred [Clearing conditions] • Cleared by reading TFX1 = 1, then writing 0 to TFX1 • Cleared by executing EP5-FIFO empty DTC transfer end interrupt exception handling 1: EP5-FIFO empty DTC transfer end interrupt request has occurred
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

6.4 Interrupt Sources

There are six types of interrupt sources: NMI, user break, H-UDI, IRQ, memory error, and on-chip peripheral modules. Each interrupt has a priority level (0 to 16), with 0 the lowest and 16 the highest. When set to level 0, that interrupt is masked at all times.

6.4.1 NMI Interrupt

The NMI interrupt has a priority level of 16 and is accepted at all times. NMI interrupt requests are edge-detected, and the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) selects whether the rising edge or falling edge is detected.

Though the priority level of the NMI interrupt is 16, the NMI interrupt exception handling sets the interrupt mask level bits (I3 to I0) in the status register (SR) to level 15.

6.4.2 User Break Interrupt

A user break interrupt which occurs when a break condition set in the user break controller (UBC) matches has a priority level of 15. The user break interrupt exception handling sets the I3 to I0 bits in SR to level 15. For user break interrupts, see section 7, User Break Controller (UBC).

6.4.3 H-UDI Interrupt

The user debugging interface (H-UDI) interrupt has a priority level of 15, and occurs at serial input of an H-UDI interrupt instruction. H-UDI interrupt requests are edge-detected and retained until they are accepted. The H-UDI interrupt exception handling sets the I3 to I0 bits in SR to level 15. For H-UDI interrupts, see section 31, User Debugging Interface (H-UDI).

6.4.4 IRQ Interrupts

IRQ interrupts are input from pins IRQ7 to IRQ0. For the IRQ interrupts, low-level, falling-edge, rising-edge, or both-edge detection can be selected individually for each pin by the IRQ sense select bits (IRQ71S to IRQ01S and IRQ70S to IRQ00S) in interrupt control register 1 (ICR1). The priority level can be set individually in a range from 0 to 15 for each pin by interrupt priority registers 01 and 02 (IPR01 and IPR02).

When using low-level setting for IRQ interrupts, an interrupt request signal is sent to the INTC while the IRQ7 to IRQ0 pins are low. An interrupt request signal is stopped being sent to the INTC when the IRQ7 to IRQ0 pins are driven high. The status of the interrupt requests can be checked by reading the IRQ interrupt request bits (IRQ7F to IRQ0F) in the IRQ interrupt request register (IRQRR).

When using edge-sensing for IRQ interrupts, an interrupt request is detected due to change of the IRQ7 to IRQ0 pin states, and an interrupt request signal is sent to the INTC. The result of IRQ interrupt request detection is retained until that interrupt request is accepted. Whether IRQ interrupt requests have been detected or not can be checked by reading the IRQ7F to IRQ0F bits in IRQRR. Writing 0 to these bits after reading them as 1 clears the result of IRQ interrupt request detection.

The IRQ interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted IRQ interrupt. Satisfaction of the setting condition for an individual IRQnF bit leads to the bit being set regardless of the setting of the I3 to I0 bits in SR.

6.4.5 Memory Error Interrupt

For details on the sources generating a memory error, see section 27, Flash Memory (ROM).

6.4.6 On-Chip Peripheral Module Interrupts

On-chip peripheral module interrupts are generated by the following on-chip peripheral modules:

- A/D converter (ADC)
- Controller area network (RCAN-ET)
- Direct memory access controller (DMAC)
- Compare match timer (CMT)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Ethernet Controller (EtherC, E-DMAC)
- USB function module (USB)
- Multi-function timer pulse unit 2 (MTU2)
- Multi-function timer pulse unit 2S (MTU2S)
- Port output enable 2 (POE2)
- I^2C bus interface 3 (IIC3)
- Renesas serial peripheral interface (RSPI)
- Serial communication interface (SCI)
- Serial communication interface with FIFO (SCIF)

As every source is assigned a different interrupt vector, the source does not need to be identified in the exception service routine. A priority level in a range from 0 to 15 can be set for each module by interrupt priority registers 05 to 19 (IPR05 to IPR19). The on-chip peripheral module interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted on-chip peripheral module interrupt.

6.5 Interrupt Exception Handling Vector Table and Priority

Table 6.4 lists interrupt sources and their vector numbers, vector table address offsets, and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from the vector numbers and vector table address offsets. In interrupt exception handling, the interrupt exception service routine start address is fetched from the vector table indicated by the vector table address. For details of calculation of the vector table address, see table 5.4 in section 5, Exception Handling.

The priorities of IRQ interrupts and on-chip peripheral module interrupts can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers 01, 02, and 05 to 19 (IPR01, IPR02, and IPR05 to IPR19). However, if two or more interrupts specified by the same IPR among IPR05 to IPR19 occur, the priorities are defined as shown in the IPR setting unit internal priority of table 6.4, and the priorities cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priorities indicated in table 6.4.

Table 6.4 Interrupt Exception Handling Vectors and Priorities

Interrupt Source	Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
NMI		11	H'0000002C to H'0000002F	16	—	—	High
UBC		12	H'00000030 to H'00000033	15	—	—	
H-UDI		14	H'00000038 to H'0000003B	15	—	—	
IRQ	IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPR01 (15 to 12)	—	
	IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPR01 (11 to 8)	—	
	IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPR01 (7 to 4)	—	
	IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPR01 (3 to 0)	—	
	IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPR02 (15 to 12)	—	
	IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPR02 (11 to 8)	—	
	IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPR02 (7 to 4)	—	
	IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPR02 (3 to 0)	—	
ROM, FLD	FIFE	91	H'0000016C to H'0000016F	15	—	—	
ADC	ADI0	92	H'00000170 to H'00000173	0 to 15 (0)	IPR05 (7 to 4)	—	
	ADI1	96	H'00000180 to H'00000183	0 to 15 (0)	IPR05 (3 to 0)	—	Low

Interrupt Source	Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
RCAN-ET	ERS_0	104	H'000001A0 to H'000001A3	0 to 15 (0)	IPR18 (11 to 8)	1	High
	OVR_0	105	H'000001A4 to H'000001A7	0 to 15 (0)		2	
	RM0_0, RM1_0	106	H'000001A8 to H'000001AB	0 to 15 (0)		3	
	SLE_0	107	H'000001AC to H'000001AF	0 to 15 (0)		4	
DMAC	DMAC0 DEI0	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR06 (15 to 12)	1	
	HEI0	109	H'000001B4 to H'000001B7			2	
	DMAC1 DEI1	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPR06 (11 to 8)	1	
	HEI1	113	H'000001C4 to H'000001C7			2	
	DMAC2 DEI2	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR06 (7 to 4)	1	
	HEI2	117	H'000001D4 to H'000001D7			2	
	DMAC3 DEI3	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR06 (3 to 0)	1	
	HEI3	121	H'000001E4 to H'000001E7			2	
	DMAC4 DEI4	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR07 (15 to 12)	1	
	HEI4	125	H'000001F4 to H'000001F7			2	
	DMAC5 DEI5	128	H'00000200 to H'00000203	0 to 15 (0)	IPR07 (11 to 8)	1	
	HEI5	129	H'00000204 to H'00000207			2	Low

Interrupt Source		Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit: Internal Priority	Default Priority
DMAC	DMAC6	DEI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR07 (7 to 4)	1	High
		HEI6	133	H'00000214 to H'00000217			2	
	DMAC7	DEI7	136	H'00000220 to H'00000223	0 to 15 (0)	IPR07 (3 to 0)	1	
		HEI7	137	H'00000224 to H'00000227			2	
CMT	CMI0		140	H'00000230 to H'00000233	0 to 15 (0)	IPR08 (15 to 12)	—	
	CMI1		144	H'00000240 to H'00000243	0 to 15 (0)	IPR08 (11 to 8)	—	
BSC	CMI		148	H'00000250 to H'00000253	0 to 15 (0)	IPR08 (7 to 4)	—	
USB	EP4-FIFO full DTC transfer end (USBRXI1)		150	H'00000258 to H'0000025B	0 to 15 (0)	IPR19 (15 to 12)	—	
	EP5-FIFO empty DTC transfer end (USBTXI1)		151	H'0000025C to H'0000025F	0 to 15 (0)	IPR19 (11 to 8)	—	
WDT	ITI		152	H'00000260 to H'00000263	0 to 15 (0)	IPR08 (3 to 0)	—	
EtherC, E-DMAC	EINT0		153	H'00000264 to H'00000267	0 to 15 (0)	IPR19 (7 to 4)	—	
USB	EP1-FIFO full DTC transfer end (USBRXI0)		154	H'00000268 to H'0000026B	0 to 15 (0)	IPR18 (7 to 4)	—	
	EP2-FIFO empty DTC transfer end (USBTXI0)		155	H'0000026C to H'0000026F	0 to 15 (0)	IPR18 (3 to 0)	—	Low

Interrupt Source Number			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU2_0	TGIA_0	156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (15 to 12)	1	High
		TGIB_0	157	H'00000274 to H'00000277			2	
		TGIC_0	158	H'00000278 to H'0000027B			3	
		TGID_0	159	H'0000027C to H'0000027F			4	
		TCIV_0	160	H'00000280 to H'00000283	0 to 15 (0)	IPR09 (11 to 8)	1	
		TGIE_0	161	H'00000284 to H'00000287			2	
		TGIF_0	162	H'00000288 to H'0000028B			3	
	MTU2_1	TGIA_1	164	H'00000290 to H'00000293	0 to 15 (0)	IPR09 (7 to 4)	1	
		TGIB_1	165	H'00000294 to H'00000297			2	
		TCIV_1	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPR09 (3 to 0)	1	
		TCIU_1	169	H'000002A4 to H'000002A7			2	
	MTU2_2	TGIA_2	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR10 (15 to 12)	1	
		TGIB_2	173	H'000002B4 to H'000002B7			2	
		TCIV_2	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPR10 (11 to 8)	1	
		TCIU_2	177	H'000002C4 to H'000002C7			2	Low

Interrupt Source	Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU2_3	TGIA_3	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR10 (7 to 4)	1	High
		TGIB_3	181	H'000002D4 to H'000002D7			2	
		TGIC_3	182	H'000002D8 to H'000002DB			3	
		TGID_3	183	H'000002DC to H'000002DF			4	
		TCIV_3	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR10 (3 to 0)	—	
	MTU2_4	TGIA_4	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR11 (15 to 12)	1	
		TGIB_4	189	H'000002F4 to H'000002F7			2	
		TGIC_4	190	H'000002F8 to H'000002FB			3	
		TGID_4	191	H'000002FC to H'000002FF			4	
		TCIV_4	192	H'00000300 to H'00000303	0 to 15 (0)	IPR11 (11 to 8)	—	
	MTU2_5	TGIU_5	196	H'00000310 to H'00000313	0 to 15 (0)	IPR11 (7 to 4)	1	
		TGIV_5	197	H'00000314 to H'00000317			2	
		TGIW_5	198	H'00000318 to H'0000031B			3	
POE2	OEI1		200	H'00000320 to H'00000323	0 to 15 (0)	IPR11 (3 to 0)	1	
	OEI2		201	H'00000324 to H'00000327			2	Low

Interrupt Source	Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2S	MTU2S_3	TGIA_3S	204	H'00000330 to H'00000333	0 to 15 (0)	IPR12 (15 to 12)	1	High
		TGIB_3S	205	H'00000334 to H'00000337			2	
		TGIC_3S	206	H'00000338 to H'0000033B			3	
		TGID_3S	207	H'0000033C to H'0000033F			4	
		TCIV_3S	208	H'00000340 to H'00000343	0 to 15 (0)	IPR12 (11 to 8)	—	
	MTU2S_4	TGIA_4S	212	H'00000350 to H'00000353	0 to 15 (0)	IPR12 (7 to 4)	1	
		TGIB_4S	213	H'00000354 to H'00000357			2	
		TGIC_4S	214	H'00000358 to H'0000035B			3	
		TGID_4S	215	H'0000035C to H'0000035F			4	
		TCIV_4S	216	H'00000360 to H'00000363	0 to 15 (0)	IPR12 (3 to 0)	—	
	MTU2S_5	TGIU_5S	220	H'00000370 to H'00000373	0 to 15 (0)	IPR13 (15 to 12)	1	
		TGIV_5S	221	H'00000374 to H'00000377			2	
		TGIW_5S	222	H'00000378 to H'0000037B			3	
POE2	OEI3		224	H'00000380 to H'00000383	0 to 15 (0)	IPR13 (11 to 8)	—	
USB	USI0		226	H'00000388 to H'0000038B	0 to 15 (0)	IPR18 (15 to 12)	1	
	USI1		227	H'0000038C to H'0000038F			2	Low

Interrupt Source	Number		Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
IIC3	STPI		228	H'00000390 to H'00000393	0 to 15 (0)	IPR13 (7 to 4)	1	High
	NAKI		229	H'00000394 to H'00000397			2	
	RXI		230	H'00000398 to H'0000039B			3	
	TXI		231	H'0000039C to H'0000039F			4	
	TEI		232	H'000003A0 to H'000003A3			5	
RSPI	SPEI		233	H'000003A4 to H'000003A7	0 to 15 (0)	IPR17 (15 to 12)	1	
	SPRI		234	H'000003A8 to H'000003AB			2	
	SPTI		235	H'000003AC to H'000003AF			3	
SCI	SCI4	ERI4	236	H'000003B0 to H'000003B3	0 to 15 (0)	IPR17 (11 to 8)	1	
		RXI4	237	H'000003B4 to H'000003B7			2	
		TXI4	238	H'000003B8 to H'000003BB			3	
		TEI4	239	H'000003BC to H'000003BF			4	
	SCI0	ERI0	240	H'000003C0 to H'000003C3	0 to 15 (0)	IPR16 (15 to 12)	1	
		RXI0	241	H'000003C4 to H'000003C7			2	
		TXI0	242	H'000003C8 to H'000003CB			3	
		TEI0	243	H'000003CC to H'000003CF			4	Low

Interrupt Source		Number	Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
SCI	SCI1	ERI1	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR16 (11 to 8)	1	High
		RXI1	245	H'000003D4 to H'000003D7			2	
		TXI1	246	H'000003D8 to H'000003DB			3	
		TEI1	247	H'000003DC to H'000003DF			4	
	SCI2	ERI2	248	H'000003E0 to H'000003E3	0 to 15 (0)	IPR16 (7 to 4)	1	
		RXI2	249	H'000003E4 to H'000003E7			2	
		TXI2	250	H'000003E8 to H'000003EB			3	
		TEI2	251	H'000003EC to H'000003EF			4	
SCIF	SCIF3	BRI3	252	H'000003F0 to H'000003F3	0 to 15 (0)	IPR14 (3 to 0)	1	
		ERI3	253	H'000003F4 to H'000003F7			2	
		RXI3	254	H'000003F8 to H'000003FB			3	
		TXI3	255	H'000003FC to H'000003FF			4	Low

6.6 Operation

6.6.1 Interrupt Operation Sequence

The sequence of interrupt operations is described below. Figure 6.2 shows the operation flow.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest-priority interrupt from the interrupt requests sent, following the priority levels set in interrupt priority registers 01, 02, and 05 to 19 (IPR01, IPR02, and IPR05 to IPR19). Lower priority interrupts are ignored*. If two of these interrupts have the same priority level or if multiple interrupts occur within a single IPR, the interrupt with the highest priority is selected, according to the default priority and IPR setting unit internal priority shown in table 6.4.
3. The priority level of the interrupt selected by the interrupt controller is compared with the interrupt level mask bits (I3 to I0) in the status register (SR) of the CPU. If the interrupt request priority level is equal to or less than the level set in bits I3 to I0, the interrupt request is ignored. If the interrupt request priority level is higher than the level in bits I3 to I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the $\overline{\text{IRQOUT}}$ pin.
5. The CPU detects the interrupt request sent from the interrupt controller when the CPU decodes the instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception handling (figure 6.4).
6. The interrupt exception service routine start address is fetched from the exception handling vector table corresponding to the accepted interrupt.
7. The status register (SR) is saved onto the stack, and the priority level of the accepted interrupt is copied to bits I3 to I0 in SR.
8. The program counter (PC) is saved onto the stack.
9. The CPU jumps to the fetched interrupt exception service routine start address and starts executing the program. The jump that occurs is not a delayed branch.
10. A high level is output from the $\overline{\text{IRQOUT}}$ pin. However, if the interrupt controller accepts an interrupt with a higher priority than the interrupt just being accepted, the $\overline{\text{IRQOUT}}$ pin holds low level.

Notes: The interrupt source flag should be cleared in the interrupt handler. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" shown in table 6.5 is required before the interrupt source sent to the CPU is actually cancelled. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then execute an RTE instruction.

* Interrupt requests that are designated as edge-sensing are held pending until the interrupt requests are accepted. IRQ interrupts, however, can be cancelled by accessing the IRQ interrupt request register (IRQRR). For details, see section 6.4.4, IRQ Interrupts.
  Interrupts held pending due to edge-sensing are cleared by a power-on reset.

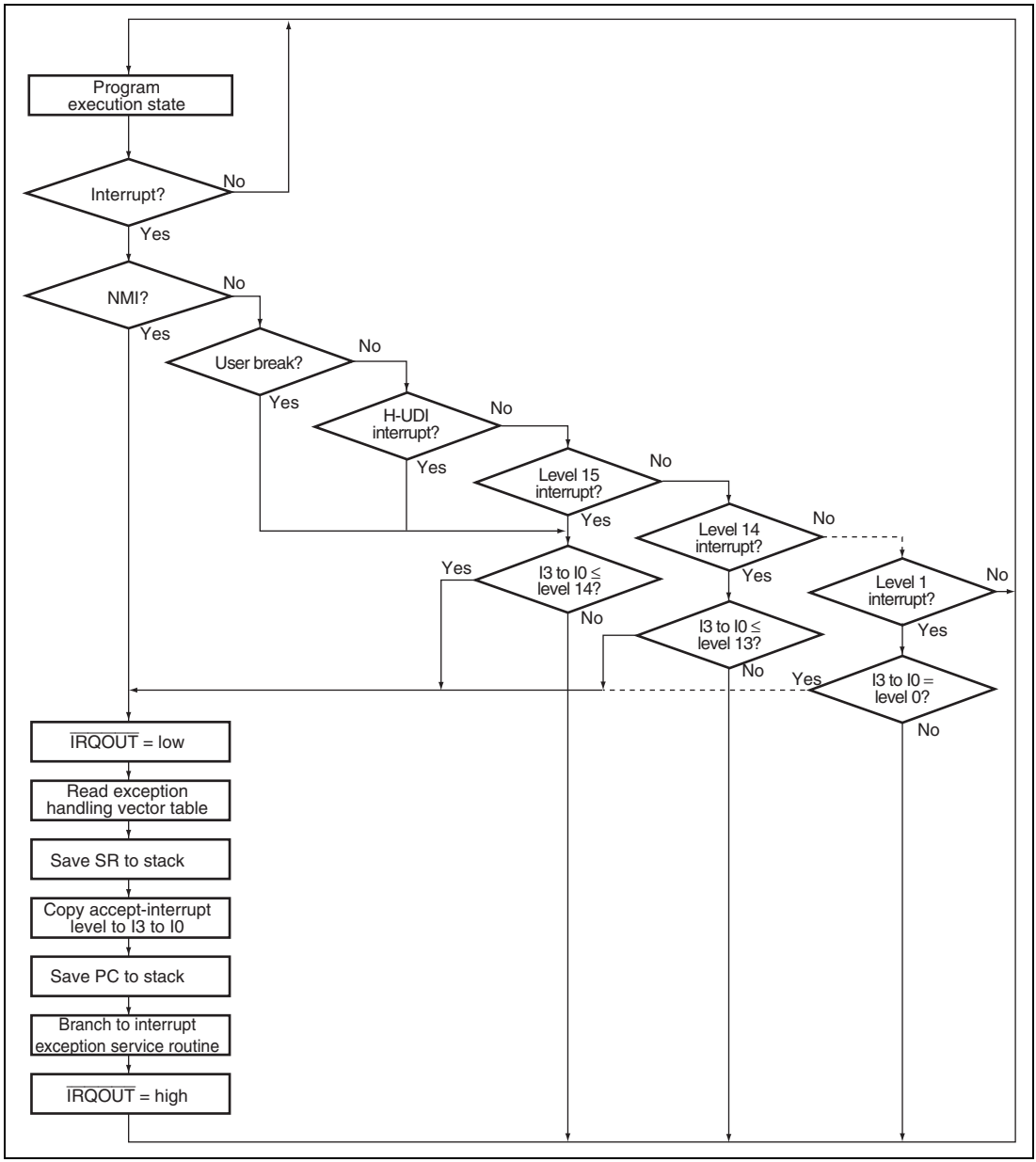
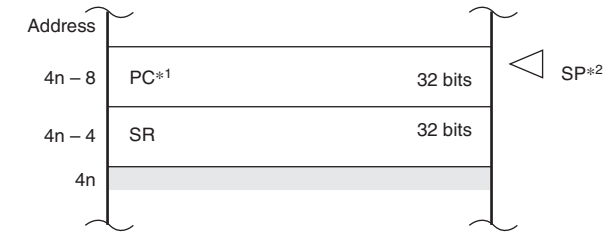


Figure 6.2 Interrupt Operation Flow

6.6.2 Stack after Interrupt Exception Handling

Figure 6.3 shows the stack after interrupt exception handling.



Notes: 1. PC: Start address of the next instruction (return destination instruction) after the executed instruction
2. Always make sure that SP is a multiple of 4.

Figure 6.3 Stack after Interrupt Exception Handling

6.7 Interrupt Response Time

Table 6.5 lists the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception handling starts and fetching of the first instruction in the exception service routine begins. The interrupt processing operations differ in the cases when banking is disabled, when banking is enabled without register bank overflow, and when banking is enabled with register bank overflow. Figures 6.4 and 6.5 show examples of pipeline operation when banking is disabled. Figures 6.6 and 6.7 show examples of pipeline operation when banking is enabled without register bank overflow. Figures 6.8 and 6.9 show examples of pipeline operation when banking is enabled with register bank overflow.

Table 6.5 Interrupt Response Time

			Number of States					
Item			**NMI**	**UBC**	**H-UDI**	**IRQ**	**Peripheral Module**	**Remarks**
Time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU			2 Icyc + 2 Bcyc + 1 Pcyc	3 Icyc	2 Icyc + 1 Pcyc	2 Icyc + 3 Bcyc + 1 Pcyc	2 Icyc + 1 Bcyc + 2 Pcyc	Interrupts with the DTC activation sources
							2 Icyc + 1 Bcyc + 1 Bcyc	Interrupts without the DTC activation sources.
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in exception service routine is fetched	No register banking	Min.	3 Icyc + m1 + m2					Min. is when the interrupt wait time is zero. Max. is when a higher-priority interrupt request has occurred during interrupt exception handling.
		Max.	4 Icyc + 2(m1 + m2) + m3					
	Register banking without register bank overflow	Min.	—	—	3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	—	12 Icyc + m1 + m2			
	Register banking with register bank overflow	Min.	—	—	3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—	—	3 Icyc + m1 + m2 + 19(m4)			

Item			Number of States					Remarks
			NMI	UBC	H-UDI	IRQ	Peripheral Module	
Interrupt response time	No register banking	Min.	5 Icyc + 2 Bcyc + 1 Pcyc + m1 + m2	6 Icyc + m1 + m2	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	• 100-MHz operation*1*2: 0.080 to 0.150 µs • 200-MHz operation*1*3: 0.040 to 0.115 µs
		Max.	6 Icyc + 2 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	7 Icyc + 2(m1 + m2) + m3	6 Icyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 3 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	6 Icyc + 1 Bcyc + 1 Pcyc + 2(m1 + m2) + m3	• 100-MHz operation*1*2: 0.120 to 0.190 µs • 200-MHz operation*1*3: 0.060 to 0.135 µs
	Register banking without register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	• 100-MHz operation*1*2: 0.080 to 0.150 µs • 200-MHz operation*1*3: 0.055 to 0.115 µs
		Max.	—	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	• 100-MHz operation*1*2: 0.170 to 0.240 µs • 200-MHz operation*1*3: 0.100 to 0.160 µs
	Register banking with register bank overflow	Min.	—	—	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	• 100-MHz operation*1*2: 0.080 to 0.150 µs • 200-MHz operation*1*3: 0.055 to 0.115 µs
		Max.	—	—	5 Icyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2 + 19(m4)	• 100-MHz operation*1*2: 0.270 to 0.340 µs • 200-MHz operation*1*3: 0.150 to 0.210 µs

Notes: m1 to m4 are the number of states needed for the following memory accesses.

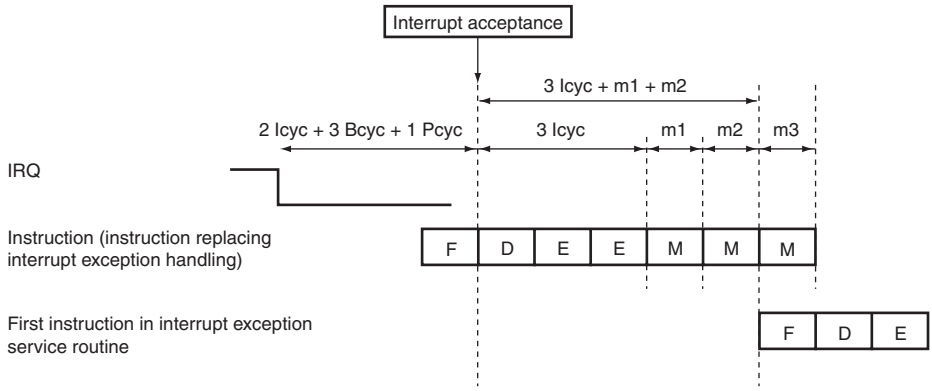
m1: Vector address read (longword read)

m2: SR save (longword write)

m3: PC save (longword write)

m4: Banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack.

1. In the case that m1 = m2 = m3 = m4 = 1 Icyc.
2. In the case that (Iϕ, Bϕ, Pϕ) = (100 MHz, 50 MHz, 50 MHz).
3. In the case that (Iϕ, Bϕ, Pϕ) = (200 MHz, 50 MHz, 50 MHz).



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)
F: Instruction fetch. Instruction is fetched from memory in which program is stored.
D: Instruction decoding. Fetched instruction is decoded.
E: Instruction execution. Data operation or address calculation is performed in accordance with the result of decoding.
M: Memory access. Memory data access is performed.

Figure 6.4 Example of Pipeline Operation when IRQ Interrupt is Accepted (No Register Banking)

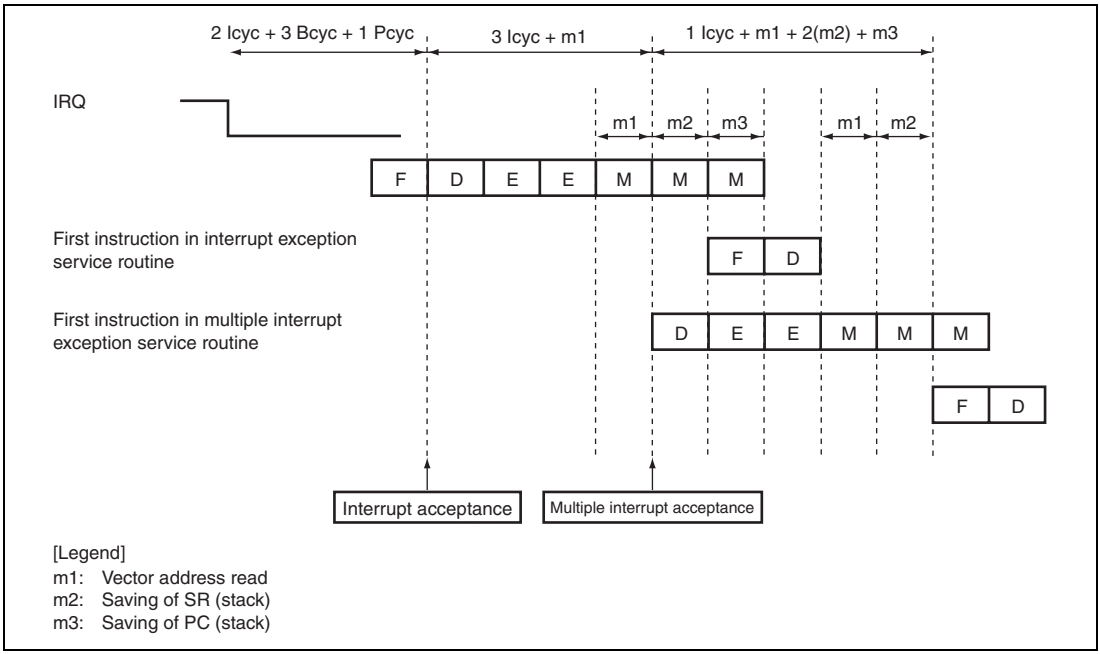


Figure 6.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

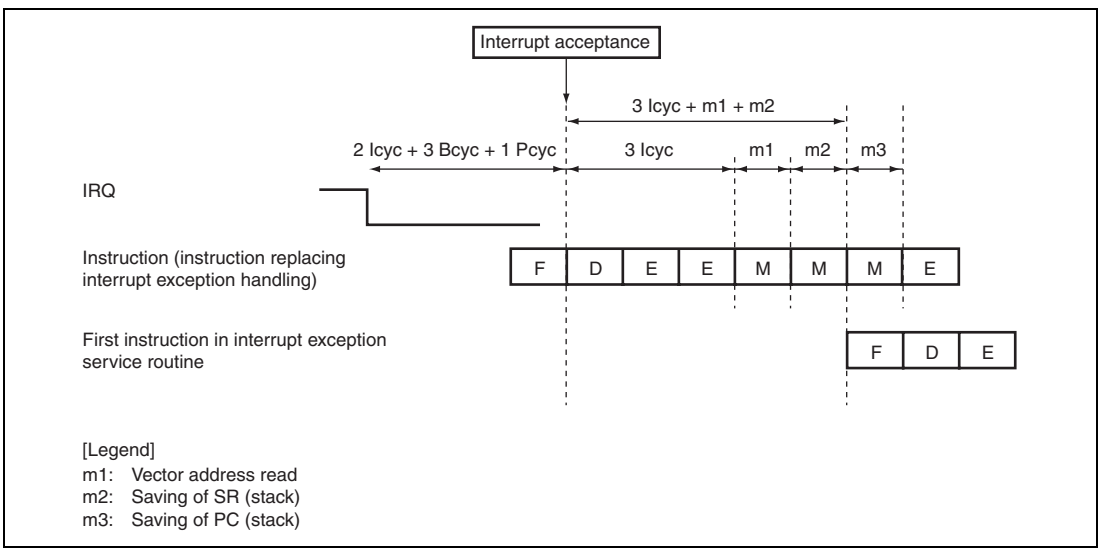


Figure 6.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

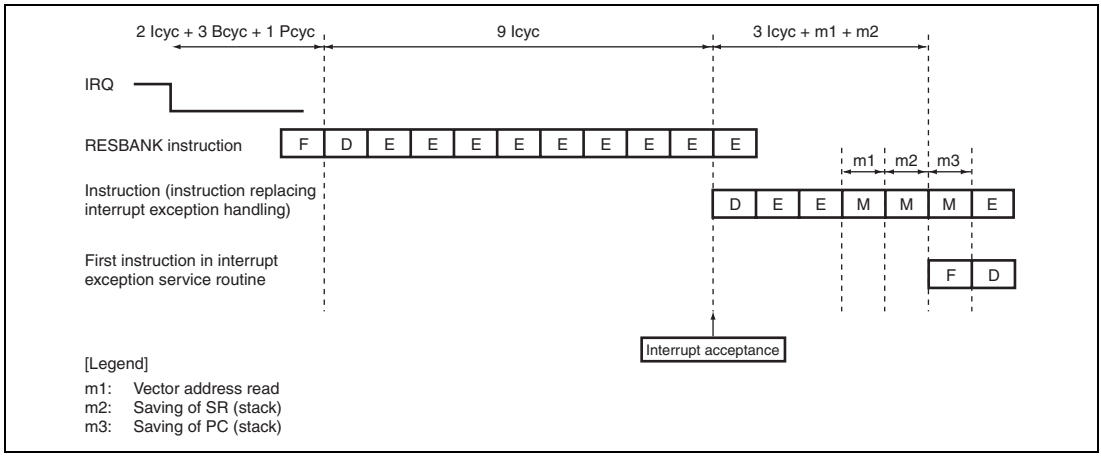


Figure 6.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

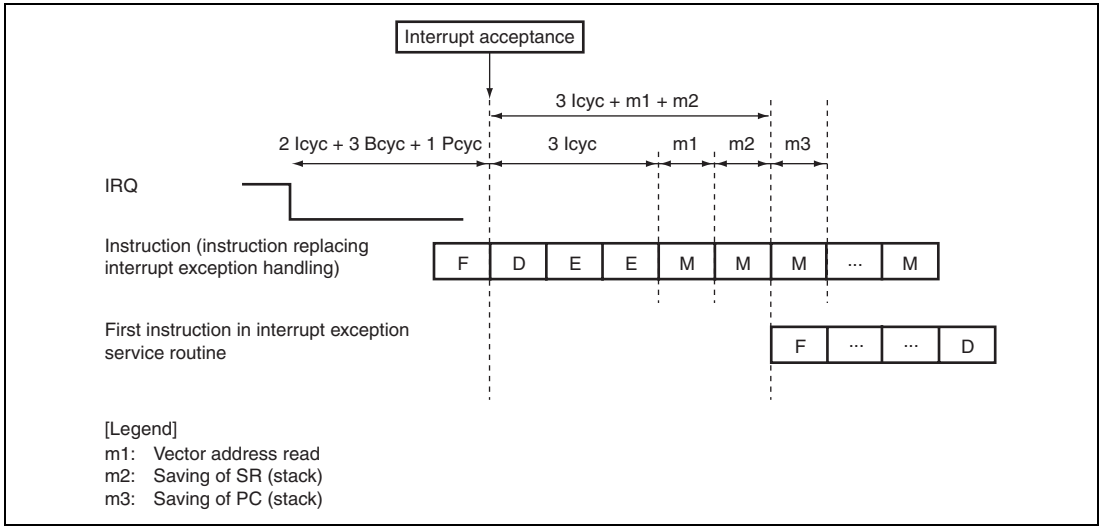


Figure 6.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

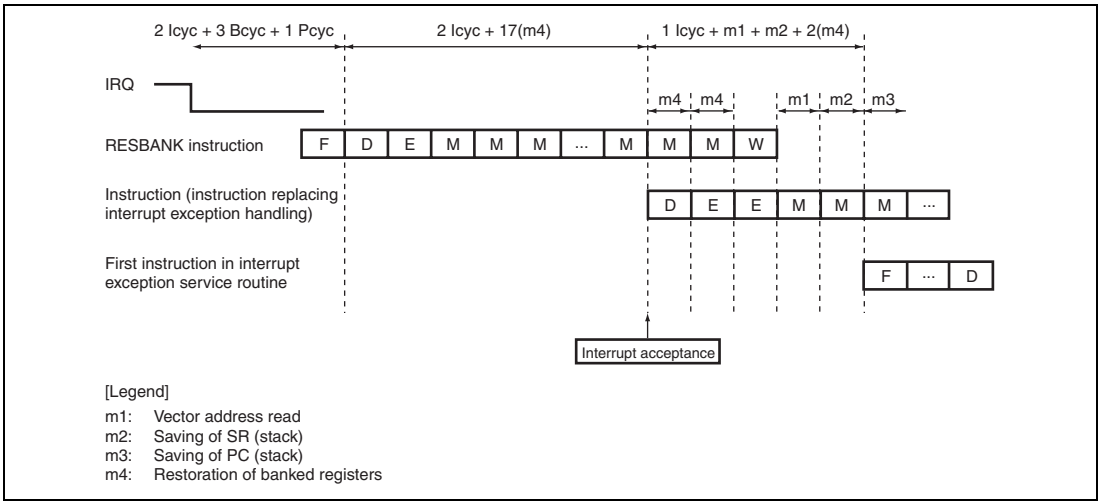


Figure 6.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

6.8 Register Banks

This LSI has fifteen register banks used to perform register saving and restoration required in the interrupt processing at high speed. Figure 6.10 shows the register bank configuration.

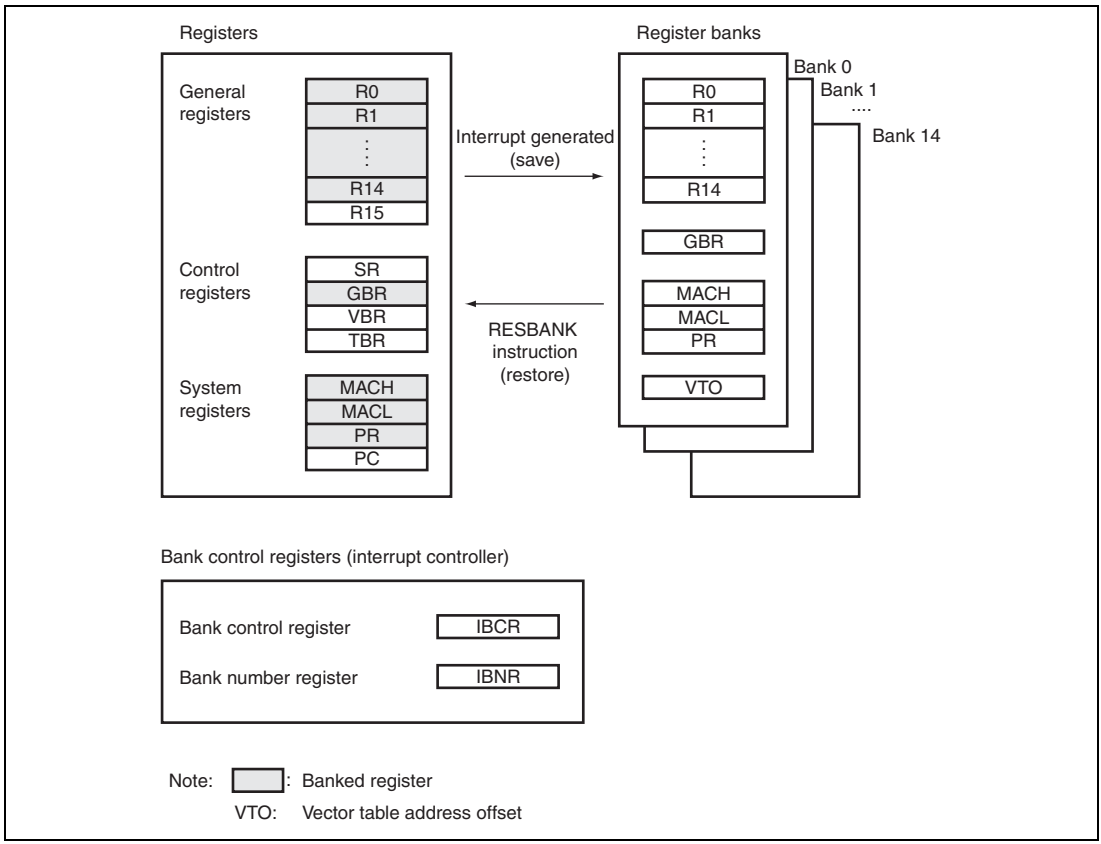


Figure 6.10 Overview of Register Bank Configuration

6.8.1 Banked Register and Input/Output of Banks

(1) Banked Register

The contents of the general registers (R0 to R14), global base register (GBR), multiply and accumulate registers (MACH and MACL), and procedure register (PR), and the vector table address offset are banked.

(2) Register Banks

This LSI has fifteen register banks, bank 0 to bank 14. Register banks are stacked in first-in last-out (FILO) sequence. Saving takes place in order, beginning from bank 0, and restoration takes place in the reverse order, beginning from the last bank saved to.

6.8.2 Bank Save and Restore Operations

(1) Saving to Bank

Figure 6.11 shows register bank save operations. The following operations are performed when an interrupt for which usage of register banks is allowed is accepted by the CPU:

a. Assume that the bank number bit value in the bank number register (IBNR), BN, is "i" before the interrupt is generated.
b. The contents of registers R0 to R14, GBR, MACH, MACL, and PR, and the interrupt vector table address offset (VTO) of the accepted interrupt are saved in the bank indicated by BN, bank i.
c. The BN value is incremented by 1.

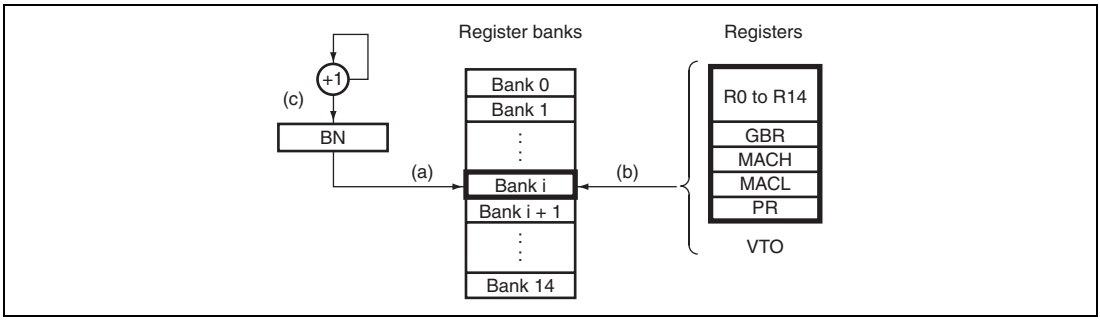


Figure 6.11 Bank Save Operations

Figure 6.12 shows the timing for saving to a register bank. Saving to a register bank takes place between the start of interrupt exception handling and the start of fetching the first instruction in the interrupt exception service routine.

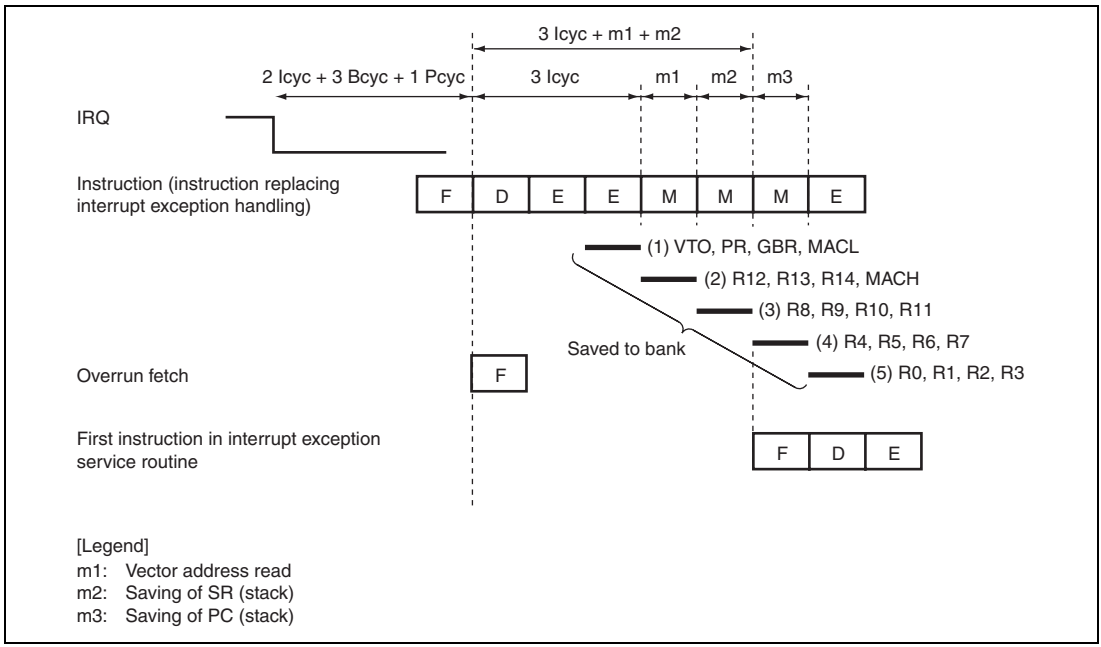


Figure 6.12 Bank Save Timing

(2) Restoration from Bank

The RESBANK (restore from register bank) instruction is used to restore data saved in a register bank. After restoring data from the register banks with the RESBANK instruction at the end of the interrupt service routine, execute the RTE instruction to return from the exception handling.

6.8.3 Save and Restore Operations after Saving to All Banks

If an interrupt occurs and usage of the register banks is enabled for the interrupt accepted by the CPU in a state where saving has been performed to all register banks, automatic saving to the stack is performed instead of register bank saving if the BOVE bit in the bank number register (IBNR) is cleared to 0. If the BOVE bit in IBNR is set to 1, register bank overflow exception occurs and data is not saved to the stack.

Save and restore operations when using the stack are as follows:

(1) Saving to Stack

1. The status register (SR) and program counter (PC) are saved to the stack during interrupt exception handling.
2. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are saved to the stack. The registers are saved to the stack in the order of MACL, MACH, GBR, PR, R14, R13, …, R1, and R0.
3. The register bank overflow bit (BO) in SR is set to 1.
4. The bank number bit (BN) value in the bank number register (IBNR) remains set to the maximum value of 15.

(2) Restoration from Stack

When the RESBANK (restore from register bank) instruction is executed with the register bank overflow bit (BO) in SR set to 1, the CPU operates as follows:

1. The contents of the banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack. The registers are restored from the stack in the order of R0, R1, …, R13, R14, PR, GBR, MACH, and MACL.
2. The bank number bit (BN) value in the bank number register (IBNR) remains set to the maximum value of 15.

6.8.4 Register Bank Exception

There are two register bank exceptions (register bank errors): register bank overflow and register bank underflow.

(1) Register Bank Overflow

This exception occurs if, after data has been saved to all of the register banks, an interrupt for which register bank use is allowed is accepted by the CPU, and the BOVE bit in the bank number register (IBNR) is set to 1. In this case, the bank number bit (BN) value in the bank number register (IBNR) remains set to the bank count of 15 and saving is not performed to the register bank.

(2) Register Bank Underflow

This exception occurs if the RESBANK (restore from register bank) instruction is executed when no data has been saved to the register banks. In this case, the values of R0 to R14, GBR, MACH, MACL, and PR do not change. In addition, the bank number bit (BN) value in the bank number register (IBNR) remains set to 0.

6.8.5 Register Bank Error Exception Handling

When a register bank error occurs, register bank error exception handling starts. When this happens, the CPU operates as follows:

1. The exception service routine start address which corresponds to the register bank error that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction for a register bank overflow, and the start address of the executed RESBANK instruction for a register bank underflow. To prevent multiple interrupts from occurring at a register bank overflow, the interrupt priority level that caused the register bank overflow is written to the interrupt mask level bits (I3 to I0) of the status register (SR).
4. Program execution starts from the exception service routine start address.

6.9 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to trigger the following data transfer.

- Only the DMAC is activated and no CPU interrupt occurs.
- Only the DTC is activated and a CPU interrupt may occur depending on the DTC setting.

Interrupt sources that are designated to activate the DMAC are masked without being input to the INTC. The mask condition is as follows:

Mask condition = DME • (DE0 • interrupt source select 0 + DE1 • interrupt source select 1 + DE2 • interrupt source select 2 + DE3 • interrupt source select 3 + DE4 • interrupt source select 4 + DE5 • interrupt source select 5 + DE6 • interrupt source select 6 + DE7 • interrupt source select 7)

Here, DME is bit 0 in DMAOR of the DMAC, and DEn (n = 0 to 7) is bit 0 in CHCR0 to CHCR7 of the DMAC. For details, see section 10, Direct Memory Access Controller (DMAC).

The INTC masks a CPU interrupt when the corresponding DTCE bit is 1. The DTCE clearing condition and interrupt source flag clearing condition are as follows:

DTCE clearing condition = DTC transfer end • DTCECLR

Interrupt source flag clearing condition = DTC transfer end • $\overline{\text{DTCECLR}}$ + DMAC transfer end

However, DTCECLR = DISEL + counter value of 0

Figures 6.13 and 6.14 show block diagrams of interrupt control.

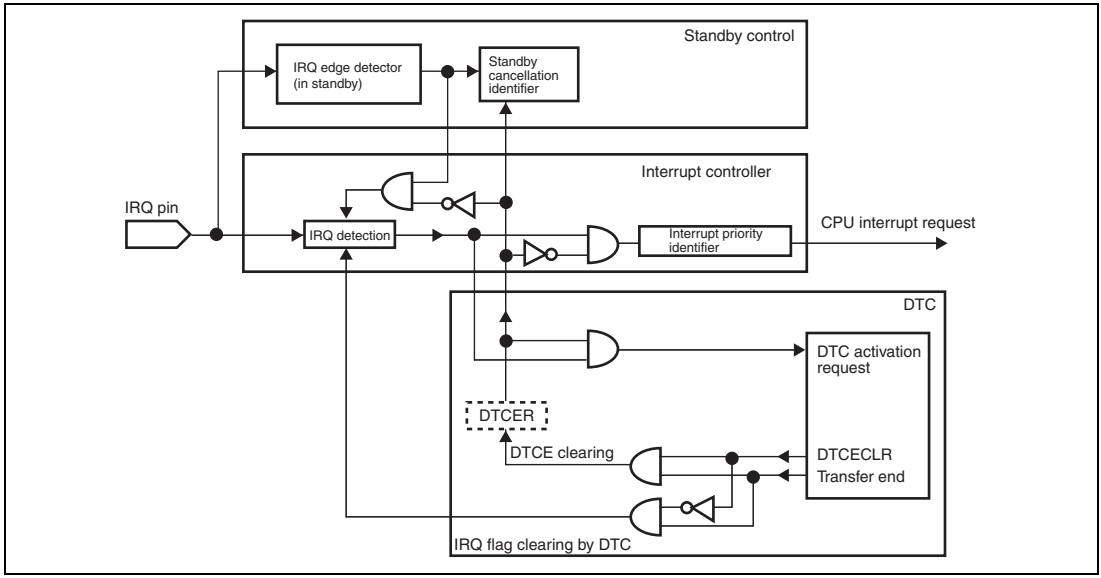


Figure 6.13 Interrupt Control Block Diagram

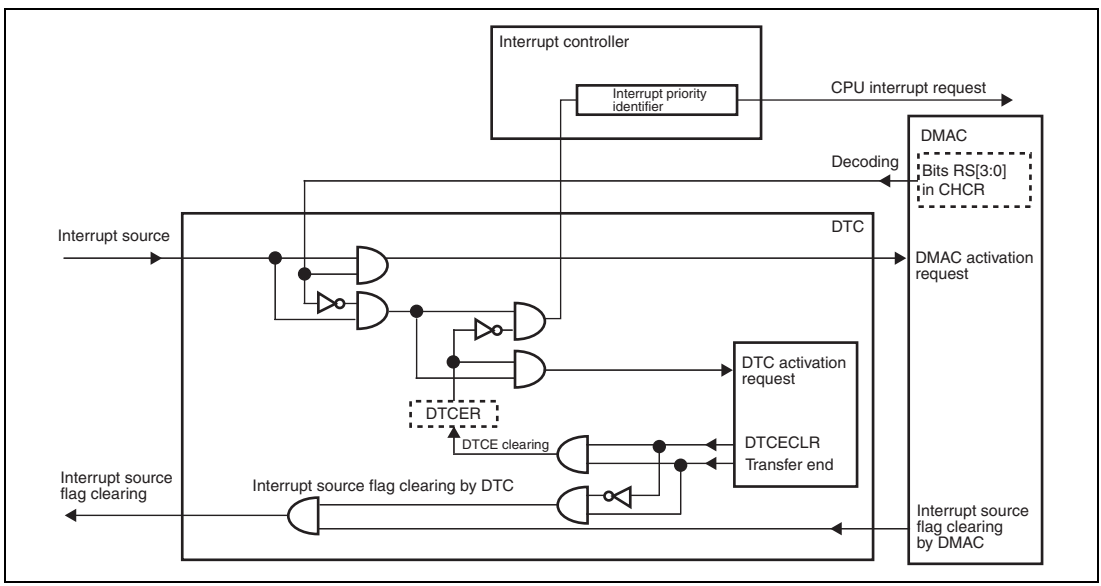


Figure 6.14 Block Diagram of Controlling an On-Chip Peripheral Module Interrupt

6.9.1 Handling Interrupt Request Signals as DTC Activating Sources and CPU Interrupt Sources but Not as DMAC Activating Sources

1. Do not select DMAC activating sources or clear the DME bit to 0. If, DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. Set both the corresponding DTCE bit and DISEL bit to 1 in the DTC.
3. Activating sources are applied to the DTC when interrupts occur.
4. The DTC clears the DTCE bit to 0 and sends interrupt requests to the CPU when starting data transfer. The DTC does not clear the activating sources.
5. The CPU clears the interrupt sources in the interrupt exception handling routine, and then confirms the transfer counter value. If the transfer counter value is not 0, the DTCE bit is set to 1 and the next data transfer enabled. If the transfer counter value is 0, the CPU performs the necessary termination processing in the interrupt exception handling routine.

6.9.2 Handling Interrupt Request Signals as DMAC Activating Sources but Not as CPU Interrupt Sources

1. Select DMAC activating sources and set both the DE and DME bits to 1. This masks CPU interrupt sources regardless of the interrupt priority register and DTC register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears the activating sources when starting data transfer.

6.9.3 Handling Interrupt Request Signals as DTC Activating Sources but Not as CPU Interrupt Sources or DMAC Activating Sources

1. Do not select DMAC activating sources or clear the DME bit to 0. If, DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. Set the corresponding DTCE bit to 1 and clear the DISEL bit to 0 in the DTC.
3. Activating sources are applied to the DTC when interrupts occur.
4. The DTC clears the activating sources when starting data transfer. Interrupt requests are not sent to the CPU because the DTCE bit remains set to 1.
5. However, when the transfer counter value is 0, the DTCE bit is cleared to 0 and interrupt requests are sent to the CPU.
6. The CPU performs the necessary termination processing in the interrupt exception handling routine.

6.9.4 Handling Interrupt Request Signals as CPU Interrupt Sources but Not as DTC Activating Sources or DMAC Activating Sources

1. Do not select DMAC activating sources or clear the DME bit to 0. If, DMAC activating sources are selected, clear the DE bit to 0 for the relevant channel of the DMAC.
2. Clear the corresponding DTCE bit to 0 in the DTC.
3. Interrupt requests are sent to the CPU when interrupts occur.
4. The CPU clears the interrupt sources and performs the necessary termination processing in the interrupt exception handling routine.

6.10 Usage Notes

6.10.1 Timing to Clear an Interrupt Source

The interrupt source flags should be cleared in the interrupt exception service routine. After clearing the interrupt source flag, "time from occurrence of interrupt request until interrupt controller identifies priority, compares it with mask bits in SR, and sends interrupt request signal to CPU" shown in table 6.5 is required before the interrupt source sent to the CPU is actually cancelled. To ensure that an interrupt request that should have been cleared is not inadvertently accepted again, read the interrupt source flag after it has been cleared, and then execute an RTE instruction.

6.10.2 In Case the NMI Pin is not in Use

When the NMI pin is not in use, fix the pin to the high level by connecting the pin to $V_{CC}Q$ via a resistor.

6.10.3 Negate Timing of $\overline{\text{IRQOUT}}$

When the interrupt controller accepts an interrupt request, a low-level signal is output from the $\overline{\text{IRQOUT}}$ pin, and after jumping to the start address of the interrupt exception service routine, a high-level signal is output from the $\overline{\text{IRQOUT}}$ pin.

However, in the case where an interrupt request is accepted by the interrupt controller and a low-level signal is output from the $\overline{\text{IRQOUT}}$ pin, but the interrupt request is canceled before a jump is made to the start address of the interrupt exception service routine, a low-level signal will be output from the $\overline{\text{IRQOUT}}$ pin until a jump is made to the start address of the interrupt exception service routine called by the next interrupt request.

6.10.4 Notes on Canceling Software Standby Mode with an IRQx Interrupt Request

When canceling software standby mode using an IRQx interrupt request, change the IRQ sense select setting of ICRx in a state in which no IRQx interrupt requests are generated and clear the IRQxF flag in IRQRRx to 0 by the automatic clearing function of the IRQx interrupt processing.

If the IRQxF flag in the IRQ interrupt request register x (IRQRRx) is 1, changing the setting of the IRQ sense select bits in the interrupt control register x (ICRx) or clearing the IRQxF flag in IRQRRx to 0 will clear the relevant IRQx interrupt request but will not clear the software standby cancellation request.

Section 7 User Break Controller (UBC)

The user break controller (UBC) provides functions that simplify program debugging. These functions make it easy to design an effective self-monitoring debugger, enabling the chip to debug programs without using an in-circuit emulator. Instruction fetch or data read/write (bus master (CPU, DMAC, or DTC) selection in the case of data read/write), data size, data contents, address value, and stop timing in the case of instruction fetch are break conditions that can be set in the UBC. Since this LSI uses a Harvard architecture, instruction fetch on the CPU bus (C bus) is performed by issuing bus cycles on the instruction fetch bus (F bus), and data access on the C bus is performed by issuing bus cycles on the memory access bus (M bus). The UBC monitors the C bus and internal bus (I bus).

7.1 Features

1. The following break comparison conditions can be set.
   Number of break channels: four channels (channels 0 to 3)
   User break can be requested as the independent condition on channels 0, 1, 2, and 3.
- Address
  Comparison of the 32-bit address is maskable in 1-bit units.
  One of the three address buses (F address bus (FAB), M address bus (MAB), and I address bus (IAB)) can be selected.
- Bus master when I bus is selected
  Selection of CPU cycles, DMAC cycles, or DTC cycles
- Bus cycle
  Instruction fetch (only when C bus is selected) or data access
- Read/write
- Operand size
  Byte, word, and longword

2. Exception handling routine for user-specified break conditions can be executed.
3. In an instruction fetch cycle, it can be selected whether PC breaks are set before or after an instruction is executed.
4. When a break condition is satisfied, a trigger signal is output from the $\overline{\text{UBCTRG}}$ pin.

Figure 7.1 shows a block diagram of the UBC.

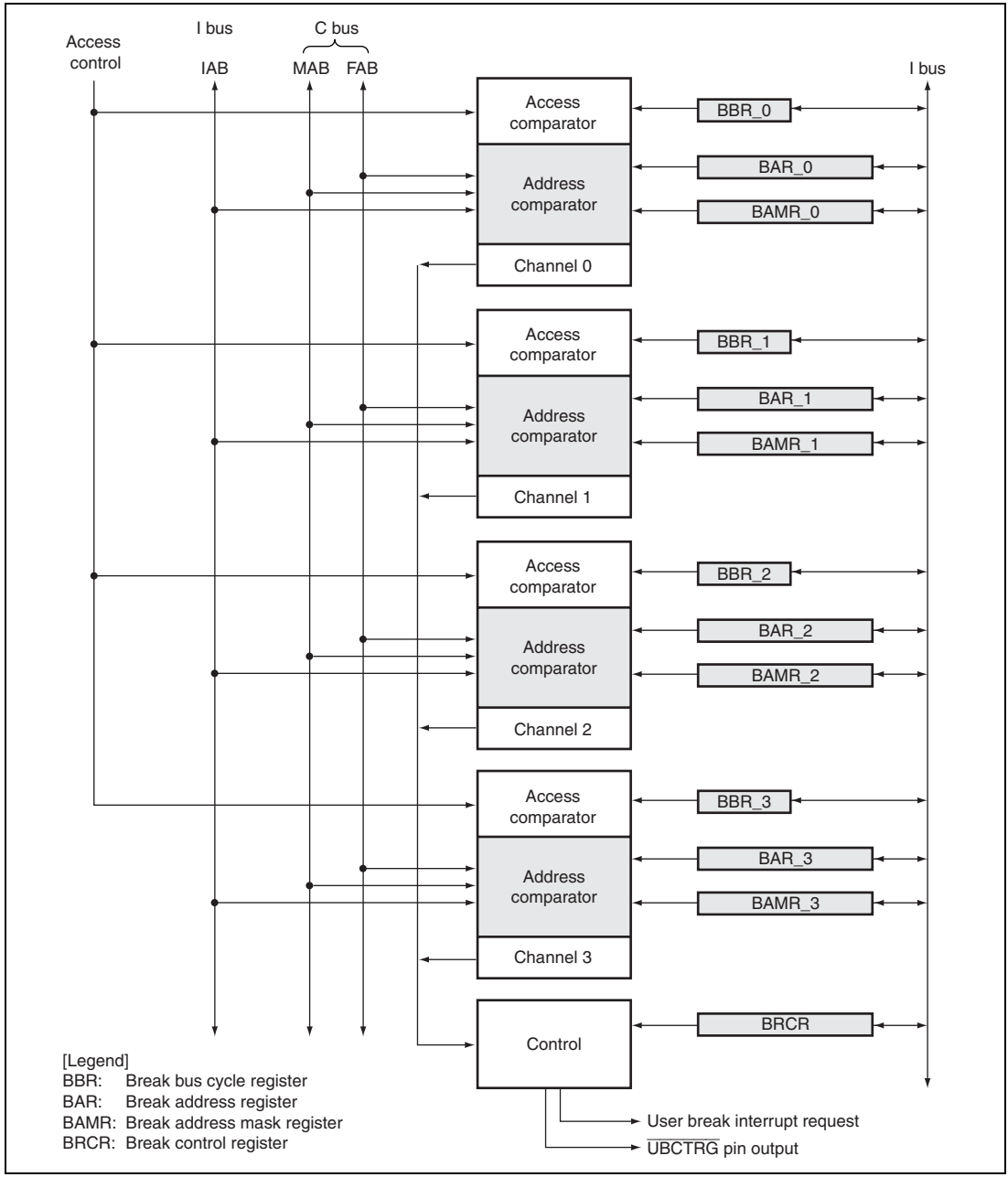


Figure 7.1 Block Diagram of UBC

7.2 Input/Output Pin

Table 7.1 shows the pin configuration of the UBC.

Table 7.1 Pin Configuration

Pin Name	Symbol	I/O	Function
UBC trigger	$\overline{\text{UBCTRG}}$	Output	Indicates that a setting condition is satisfied on either channel 0, 1, 2, or 3 of the UBC.

7.3 Register Descriptions

The UBC has the following registers.

Table 7.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Break address register_0	BAR_0	R/W	H'00000000	H'FFFC0400	32
	Break address mask register_0	BAMR_0	R/W	H'00000000	H'FFFC0404	32
	Break bus cycle register_0	BBR_0	R/W	H'0000	H'FFFC04A0	16
1	Break address register_1	BAR_1	R/W	H'00000000	H'FFFC0410	32
	Break address mask register_1	BAMR_1	R/W	H'00000000	H'FFFC0414	32
	Break bus cycle register_1	BBR_1	R/W	H'0000	H'FFFC04B0	16
2	Break address register_2	BAR_2	R/W	H'00000000	H'FFFC0420	32
	Break address mask register_2	BAMR_2	R/W	H'00000000	H'FFFC0424	32
	Break bus cycle register_2	BBR_2	R/W	H'0000	H'FFFC04A4	16
3	Break address register_3	BAR_3	R/W	H'00000000	H'FFFC0430	32
	Break address mask register_3	BAMR_3	R/W	H'00000000	H'FFFC0434	32
	Break bus cycle register_3	BBR_3	R/W	H'0000	H'FFFC04B4	16
Common	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0	32

7.3.1 Break Address Register_0 (BAR_0)

BAR_0 is a 32-bit readable/writable register. BAR_0 specifies the address used as a break condition in channel 0. The control bits CD0_1 and CD0_0 in the break bus cycle register_0 (BBR_0) select one of the three address buses for a break condition of channel 0. BAR_0 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA0_31	BA0_30	BA0_29	BA0_28	BA0_27	BA0_26	BA0_25	BA0_24	BA0_23	BA0_22	BA0_21	BA0_20	BA0_19	BA0_18	BA0_17	BA0_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA0_15	BA0_14	BA0_13	BA0_12	BA0_11	BA0_10	BA0_9	BA0_8	BA0_7	BA0_6	BA0_5	BA0_4	BA0_3	BA0_2	BA0_1	BA0_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA0_31 to BA0_0	All 0	R/W	Break Address 0 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 0. When the C bus and instruction fetch cycle are selected by BBR_0, specify an FAB address in bits BA0_31 to BA0_0. When the C bus and data access cycle are selected by BBR_0, specify an MAB address in bits BA0_31 to BA0_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR_0 to 0.

7.3.2 Break Address Mask Register_0 (BAMR_0)

BAMR_0 is a 32-bit readable/writable register. BAMR_0 specifies bits masked in the break address bits specified by BAR_0. BAMR_0 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM0_31	BAM0_30	BAM0_29	BAM0_28	BAM0_27	BAM0_26	BAM0_25	BAM0_24	BAM0_23	BAM0_22	BAM0_21	BAM0_20	BAM0_19	BAM0_18	BAM0_17	BAM0_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM0_15	BAM0_14	BAM0_13	BAM0_12	BAM0_11	BAM0_10	BAM0_9	BAM0_8	BAM0_7	BAM0_6	BAM0_5	BAM0_4	BAM0_3	BAM0_2	BAM0_1	BAM0_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM0_31 to BAM0_0	All 0	R/W	Break Address Mask 0 Specify bits masked in the channel-0 break address bits specified by BAR_0 (BA0_31 to BA0_0). 0: Break address bit BA0_n is included in the break condition 1: Break address bit BA0_n is masked and not included in the break condition

Note: n = 31 to 0

7.3.3 Break Bus Cycle Register_0 (BBR_0)

BBR_0 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) including or excluding of the data bus value, (3) bus master of the I bus, (4) C bus cycle or I bus cycle, (5) instruction fetch or data access, (6) read or write, and (7) operand size as the break conditions of channel 0. BBR_0 is initialized to H'0000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID0	-	-	CP0[2:0]			CD0[1:0]		ID0[1:0]		RW0[1:0]		SZ0[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID0	0	R/W	User Break Interrupt Disable 0 Disables or enables user break interrupt requests when a channel-0 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP0[2:0]	000	R/W	I-Bus Bus Master Select 0 Select the bus master when the bus cycle of the channel-0 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: DTC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD0[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 0 Select the C bus cycle or I bus cycle as the bus cycle of the channel-0 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID0[1:0]	00	R/W	Instruction Fetch/Data Access Select 0 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel-0 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW0[1:0]	00	R/W	Read/Write Select 0 Select the read cycle or write cycle as the bus cycle of the channel-0 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ0[1:0]	00	R/W	Operand Size Select 0 Select the operand size of the bus cycle for the channel-0 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

7.3.4 Break Address Register_1 (BAR_1)

BAR_1 is a 32-bit readable/writable register. BAR_1 specifies the address used as a break condition in channel 1. The control bits CD1_1 and CD1_0 in the break bus cycle register_1 (BBR_1) select one of the three address buses for a break condition of channel 1. BAR_1 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA1_31	BA1_30	BA1_29	BA1_28	BA1_27	BA1_26	BA1_25	BA1_24	BA1_23	BA1_22	BA1_21	BA1_20	BA1_19	BA1_18	BA1_17	BA1_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA1_15	BA1_14	BA1_13	BA1_12	BA1_11	BA1_10	BA1_9	BA1_8	BA1_7	BA1_6	BA1_5	BA1_4	BA1_3	BA1_2	BA1_1	BA1_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA1_31 to BA1_0	All 0	R/W	Break Address 1 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 1. When the C bus and instruction fetch cycle are selected by BBR_1, specify an FAB address in bits BA1_31 to BA1_0. When the C bus and data access cycle are selected by BBR_1, specify an MAB address in bits BA1_31 to BA1_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR_1 to 0.

7.3.5 Break Address Mask Register_1 (BAMR_1)

BAMR_1 is a 32-bit readable/writable register. BAMR_1 specifies bits masked in the break address bits specified by BAR_1. BAMR_1 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM1_31	BAM1_30	BAM1_29	BAM1_28	BAM1_27	BAM1_26	BAM1_25	BAM1_24	BAM1_23	BAM1_22	BAM1_21	BAM1_20	BAM1_19	BAM1_18	BAM1_17	BAM1_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM1_15	BAM1_14	BAM1_13	BAM1_12	BAM1_11	BAM1_10	BAM1_9	BAM1_8	BAM1_7	BAM1_6	BAM1_5	BAM1_4	BAM1_3	BAM1_2	BAM1_1	BAM1_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM1_31 to BAM1_0	All 0	R/W	Break Address Mask 1 Specify bits masked in the channel-1 break address bits specified by BAR_1 (BA1_31 to BA1_0). 0: Break address bit BA1_n is included in the break condition 1: Break address bit BA1_n is masked and not included in the break condition

Note: n = 31 to 0

7.3.6 Break Bus Cycle Register_1 (BBR_1)

BBR_1 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) including or excluding of the data bus value, (3) bus master of the I bus, (4) C bus cycle or I bus cycle, (5) instruction fetch or data access, (6) read or write, and (7) operand size as the break conditions of channel 1. BBR_1 is initialized to H'0000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID1	-	-	CP1[2:0]			CD1[1:0]		ID1[1:0]		RW1[1:0]		SZ1[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID1	0	R/W	User Break Interrupt Disable 1 Disables or enables user break interrupt requests when a channel-1 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP1[2:0]	000	R/W	I-Bus Bus Master Select 1 Select the bus master when the bus cycle of the channel-1 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: DTC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD1[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 1 Select the C bus cycle or I bus cycle as the bus cycle of the channel-1 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID1[1:0]	00	R/W	Instruction Fetch/Data Access Select 1 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel-1 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW1[1:0]	00	R/W	Read/Write Select 1 Select the read cycle or write cycle as the bus cycle of the channel-1 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ1[1:0]	00	R/W	Operand Size Select 1 Select the operand size of the bus cycle for the channel-1 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

7.3.7 Break Address Register_2 (BAR_2)

BAR_2 is a 32-bit readable/writable register. BAR_2 specifies the address used as a break condition in channel 2. The control bits CD2_1 and CD2_0 in the break bus cycle register_2 (BBR_2) select one of the three address buses for a break condition of channel 2. BAR_2 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA2_31	BA2_30	BA2_29	BA2_28	BA2_27	BA2_26	BA2_25	BA2_24	BA2_23	BA2_22	BA2_21	BA2_20	BA2_19	BA2_18	BA2_17	BA2_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA2_15	BA2_14	BA2_13	BA2_12	BA2_11	BA2_10	BA2_9	BA2_8	BA2_7	BA2_6	BA2_5	BA2_4	BA2_3	BA2_2	BA2_1	BA2_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA2_31 to BA2_0	All 0	R/W	Break Address 2 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 2. When the C bus and instruction fetch cycle are selected by BBR_2, specify an FAB address in bits BA2_31 to BA2_0. When the C bus and data access cycle are selected by BBR_2, specify an MAB address in bits BA2_31 to BA0_2.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR_2 to 0.

7.3.8 Break Address Mask Register_2 (BAMR_2)

BAMR_2 is a 32-bit readable/writable register. BAMR_2 specifies bits masked in the break address bits specified by BAR_2. BAMR_2 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM2_31	BAM2_30	BAM2_29	BAM2_28	BAM2_27	BAM2_26	BAM2_25	BAM2_24	BAM2_23	BAM2_22	BAM2_21	BAM2_20	BAM2_19	BAM2_18	BAM2_17	BAM2_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM2_15	BAM2_14	BAM2_13	BAM2_12	BAM2_11	BAM2_10	BAM2_9	BAM2_8	BAM2_7	BAM2_6	BAM2_5	BAM2_4	BAM2_3	BAM2_2	BAM2_1	BAM2_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM2_31 to BAM2_0	All 0	R/W	Break Address Mask 2 Specify bits masked in the channel-2 break address bits specified by BAR_2 (BA2_31 to BA2_0). 0: Break address bit BA2_n is included in the break condition 1: Break address bit BA2_n is masked and not included in the break condition

Note: n = 31 to 0

7.3.9 Break Bus Cycle Register_2 (BBR_2)

BBR_2 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) including or excluding of the data bus value, (3) bus master of the I bus, (4) C bus cycle or I bus cycle, (5) instruction fetch or data access, (6) read or write, and (7) operand size as the break conditions of channel 2. BBR_2 is initialized to H'0000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID2	-	-	CP2[2:0]			CD2[1:0]		ID2[1:0]		RW2[1:0]		SZ2[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID2	0	R/W	User Break Interrupt Disable 2 Disables or enables user break interrupt requests when a channel-2 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP2[2:0]	000	R/W	I-Bus Bus Master Select 2 Select the bus master when the bus cycle of the channel-2 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: DTC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD2[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 2 Select the C bus cycle or I bus cycle as the bus cycle of the channel-2 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID2[1:0]	00	R/W	Instruction Fetch/Data Access Select 2 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel-2 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW2[1:0]	00	R/W	Read/Write Select 2 Select the read cycle or write cycle as the bus cycle of the channel-2 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ2[1:0]	00	R/W	Operand Size Select 2 Select the operand size of the bus cycle for the channel-2 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

7.3.10 Break Address Register_3 (BAR_3)

BAR_3 is a 32-bit readable/writable register. BAR_3 specifies the address used as a break condition in channel 3. The control bits CD3_1 and CD3_0 in the break bus cycle register_3 (BBR_3) select one of the three address buses for a break condition of channel 3. BAR_3 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA3_31	BA3_30	BA3_29	BA3_28	BA3_27	BA3_26	BA3_25	BA3_24	BA3_23	BA3_22	BA3_21	BA3_20	BA3_19	BA3_18	BA3_17	BA3_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BA3_15	BA3_14	BA3_13	BA3_12	BA3_11	BA3_10	BA3_9	BA3_8	BA3_7	BA3_6	BA3_5	BA3_4	BA3_3	BA3_2	BA3_1	BA3_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BA3_31 to BA3_0	All 0	R/W	Break Address 3 Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions of channel 3. When the C bus and instruction fetch cycle are selected by BBR_3, specify an FAB address in bits BA3_31 to BA3_0. When the C bus and data access cycle are selected by BBR_3, specify an MAB address in bits BA3_31 to BA3_0.

Note: When setting the instruction fetch cycle as a break condition, clear the LSB in BAR_3 to 0.

7.3.11 Break Address Mask Register_3 (BAMR_3)

BAMR_3 is a 32-bit readable/writable register. BAMR_3 specifies bits masked in the break address bits specified by BAR_3. BAMR_3 is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM3_31	BAM3_30	BAM3_29	BAM3_28	BAM3_27	BAM3_26	BAM3_25	BAM3_24	BAM3_23	BAM3_22	BAM3_21	BAM3_20	BAM3_19	BAM3_18	BAM3_17	BAM3_16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAM3_15	BAM3_14	BAM3_13	BAM3_12	BAM3_11	BAM3_10	BAM3_9	BAM3_8	BAM3_7	BAM3_6	BAM3_5	BAM3_4	BAM3_3	BAM3_2	BAM3_1	BAM3_0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	BAM3_31 to BAM3_0	All 0	R/W	Break Address Mask 3 Specify bits masked in the channel-3 break address bits specified by BAR_3 (BA3_31 to BA3_0). 0: Break address bit BA3_n is included in the break condition 1: Break address bit BA3_n is masked and not included in the break condition

Note: n = 31 to 0

7.3.12 Break Bus Cycle Register_3 (BBR_3)

BBR_3 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrupts, (2) including or excluding of the data bus value, (3) bus master of the I bus, (4) C bus cycle or I bus cycle, (5) instruction fetch or data access, (6) read or write, and (7) operand size as the break conditions of channel 3. BBR_3 is initialized to H'0000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID3	-	-	CP3[2:0]			CD3[1:0]		ID3[1:0]		RW3[1:0]		SZ3[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	UBID3	0	R/W	User Break Interrupt Disable 3 Disables or enables user break interrupt requests when a channel-3 break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12, 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	CP3[2:0]	000	R/W	I-Bus Bus Master Select 3 Select the bus master when the bus cycle of the channel-3 break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xx1: CPU cycle is included in break conditions x1x: DMAC cycle is included in break conditions 1xx: DTC cycle is included in break conditions

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD3[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select 3 Select the C bus cycle or I bus cycle as the bus cycle of the channel-3 break condition. 00: Condition comparison is not performed 01: Break condition is the C bus (F bus or M bus) cycle 10: Break condition is the I bus cycle 11: Break condition is the C bus (F bus or M bus) cycle
5, 4	ID3[1:0]	00	R/W	Instruction Fetch/Data Access Select 3 Select the instruction fetch cycle or data access cycle as the bus cycle of the channel-3 break condition. If the instruction fetch cycle is selected, select the C bus cycle. 00: Condition comparison is not performed 01: Break condition is the instruction fetch cycle 10: Break condition is the data access cycle 11: Break condition is the instruction fetch cycle or data access cycle
3, 2	RW3[1:0]	00	R/W	Read/Write Select 3 Select the read cycle or write cycle as the bus cycle of the channel-3 break condition. 00: Condition comparison is not performed 01: Break condition is the read cycle 10: Break condition is the write cycle 11: Break condition is the read cycle or write cycle
1, 0	SZ3[1:0]	00	R/W	Operand Size Select 3 Select the operand size of the bus cycle for the channel-3 break condition. 00: Break condition does not include operand size 01: Break condition is byte access 10: Break condition is word access 11: Break condition is longword access

[Legend]

x: Don't care

7.3.13 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether user breaks are set before or after instruction execution.
2. Specifies the pulse width of the $\overline{\text{UBCTRG}}$ output when a break condition is satisfied.

BRCR is a 32-bit readable/writable register that has break condition match flags and bits for setting other break conditions. For the condition match flags of bits 15 to 12, writing 1 is invalid (previous values are retained) and writing 0 is only possible. To clear the flag, write 0 to the flag bit to be cleared and 1 to all other flag bits. BRCR is initialized to H'00000000 by a power-on reset, but retains its previous value by a manual reset or in software standby mode or sleep mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCMFC 0	SCMFC 1	SCMFC 2	SCMFC 3	SCMFD 0	SCMFD 1	SCMFD 2	SCMFD 3	PCB3	PCB2	PCB1	PCB0	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	CKS[1:0]	00	R/W	Clock Select These bits specify the pulse width output to the $\overline{\text{UBCTRG}}$ pin when a break condition is satisfied. 00: Pulse width of $\overline{\text{UBCTRG}}$ is one bus clock cycle 01: Pulse width of $\overline{\text{UBCTRG}}$ is two bus clock cycles 10: Pulse width of $\overline{\text{UBCTRG}}$ is four bus clock cycles 11: Pulse width of $\overline{\text{UBCTRG}}$ is eight bus clock cycles

Bit	Bit Name	Initial Value	R/W	Description
15	SCMFC0	0	R/W	C Bus Cycle Condition Match Flag 0 When the C bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 0 does not match 1: The C bus cycle condition for channel 0 matches
14	SCMFC1	0	R/W	C Bus Cycle Condition Match Flag 1 When the C bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 1 does not match 1: The C bus cycle condition for channel 1 matches
13	SCMFC2	0	R/W	C Bus Cycle Condition Match Flag 2 When the C bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 2 does not match 1: The C bus cycle condition for channel 2 matches
12	SCMFC3	0	R/W	C Bus Cycle Condition Match Flag 3 When the C bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The C bus cycle condition for channel 3 does not match 1: The C bus cycle condition for channel 3 matches
11	SCMFD0	0	R/W	I Bus Cycle Condition Match Flag 0 When the I bus cycle condition in the break conditions set for channel 0 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 0 does not match 1: The I bus cycle condition for channel 0 matches

Bit	Bit Name	Initial Value	R/W	Description
10	SCMFD1	0	R/W	I Bus Cycle Condition Match Flag 1 When the I bus cycle condition in the break conditions set for channel 1 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 1 does not match 1: The I bus cycle condition for channel 1 matches
9	SCMFD2	0	R/W	I Bus Cycle Condition Match Flag 2 When the I bus cycle condition in the break conditions set for channel 2 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 2 does not match 1: The I bus cycle condition for channel 2 matches
8	SCMFD3	0	R/W	I Bus Cycle Condition Match Flag 3 When the I bus cycle condition in the break conditions set for channel 3 is satisfied, this flag is set to 1. In order to clear this flag, write 0 to this bit. 0: The I bus cycle condition for channel 3 does not match 1: The I bus cycle condition for channel 3 matches
7	PCB3	0	R/W	PC Break Select 3 Selects the break timing of the instruction fetch cycle for channel 3 as before or after instruction execution. 0: PC break of channel 3 is generated before instruction execution 1: PC break of channel 3 is generated after instruction execution
6	PCB2	0	R/W	PC Break Select 2 Selects the break timing of the instruction fetch cycle for channel 2 as before or after instruction execution. 0: PC break of channel 2 is generated before instruction execution 1: PC break of channel 2 is generated after instruction execution

Bit	Bit Name	Initial Value	R/W	Description
5	PCB1	0	R/W	PC Break Select 1 Selects the break timing of the instruction fetch cycle for channel 1 as before or after instruction execution. 0: PC break of channel 1 is generated before instruction execution 1: PC break of channel 1 is generated after instruction execution
4	PCB0	0	R/W	PC Break Select 0 Selects the break timing of the instruction fetch cycle for channel 0 as before or after instruction execution. 0: PC break of channel 0 is generated before instruction execution 1: PC break of channel 0 is generated after instruction execution
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

7.4 Operation

7.4.1 Flow of the User Break Operation

The flow from setting of break conditions to user break interrupt exception handling is described below:

1. The break address is set in a break address register (BAR). The masked address bits are set in a break address mask register (BAMR). The bus break conditions are set in the break bus cycle register (BBR). Three control bit groups of BBR (C bus cycle/I bus cycle select, instruction fetch/data access select, and read/write select) are each set. No user break will be generated if even one of these groups is set to 00. The relevant break control conditions are set in the bits of the break control register (BRCR). Make sure to set all registers related to breaks before setting BBR, and branch after reading from the last written register. The newly written register values become valid from the instruction at the branch destination.
2. In the case where the break conditions are satisfied, the UBC sends a user break interrupt request to the CPU, sets the C bus condition match flag (SCMFC) or I bus condition match flag (SCMFD) for the appropriate channel, and outputs a pulse to the $\overline{\text{UBCTRG}}$ pin with the width set by the CKS1 and CKS0 bits. Setting the UBID bit in BBR to 1 enables external monitoring of the trigger output without requesting user break interrupts.
3. On receiving a user break interrupt request signal, the INTC determines its priority. Since the user break interrupt has a priority level of 15, it is accepted when the priority level set in the interrupt mask level bits (I3 to I0) of the status register (SR) is 14 or lower. If the I3 to I0 bits are set to a priority level of 15, the user break interrupt is not accepted, but the conditions are checked, and condition match flags are set if the conditions match. For details on ascertaining the priority, see section 6, Interrupt Controller (INTC).
4. Condition match flags (SCMFC and SCMFD) can be used to check which condition has been satisfied. They are set when the conditions match, but are not reset. To use these flags again, write 0 to the corresponding bit of the flags.
5. It is possible that the breaks set in channels 0 to 3 occur around the same time. In this case, there will be only one user break request to the CPU, but these four break channel match flags may be set at the same time.

6. When selecting the I bus as the break condition, note as follows:
   — Several bus masters, including the CPU and DMAC, are connected to the I bus. The UBC monitors bus cycles generated by the bus master specified by BBR, and determines the condition match.
   — I bus cycles (including read fill cycles) resulting from instruction fetches on the C bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.
   — The DTC and DMAC only issue data access cycles for I bus cycles.
   — If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the user break is to be accepted cannot be clearly defined.

7.4.2 Break on Instruction Fetch Cycle

1. When C bus/instruction fetch/read/word or longword is set in the break bus cycle register (BBR), the break condition is the FAB bus instruction fetch cycle. Whether PC breaks are set before or after the execution of the instruction can then be selected with the PCB0 or PCB1 bit of the break control register (BRCR) for the appropriate channel. If an instruction fetch cycle is set as a break condition, clear LSB in the break address register (BAR) to 0. A break cannot be generated as long as this bit is set to 1.
2. A break for instruction fetch which is set as a break before instruction execution occurs when it is confirmed that the instruction has been fetched and will be executed. This means a break does not occur for instructions fetched by overrun (instructions fetched at a branch or during an interrupt transition, but not to be executed). When this kind of break is set for the delay slot of a delayed branch instruction, the break is not generated until the execution of the first instruction at the branch destination.

   Note: If a branch does not occur at a delayed branch instruction, the subsequent instruction is not recognized as a delay slot.

3. When setting a break condition for break after instruction execution, the instruction set with the break condition is executed and then the break is generated prior to execution of the next instruction. As with pre-execution breaks, a break does not occur with overrun fetch instructions. When this kind of break is set for a delayed branch instruction and its delay slot, the break is not generated until the first instruction at the branch destination.
4. When an instruction fetch cycle is set, the break data register (BDR) is ignored. Therefore, break data cannot be set for the break of the instruction fetch cycle.
5. If the I bus is set for a break of an instruction fetch cycle, the setting is invalidated.

7.4.3 Break on Data Access Cycle

1. If the C bus is specified as a break condition for data access break, condition comparison is performed for the virtual address accessed by the executed instructions, and a break occurs if the condition is satisfied. If the I bus is specified as a break condition, condition comparison is performed for the physical address of the data access cycles that are issued by the bus master specified by the bits to select the bus master of the I bus, and a break occurs if the condition is satisfied. For details on the CPU bus cycles issued on the I bus, see 6 in section 7.4.1, Flow of the User Break Operation.
2. The relationship between the data access cycle address and the comparison condition for each operand size is listed in table 7.3.

Table 7.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

Access Size	Address Compared
Longword	Compares break address register bits 31 to 2 to address bus bits 31 to 2
Word	Compares break address register bits 31 to 1 to address bus bits 31 to 1
Byte	Compares break address register bits 31 to 0 to address bus bits 31 to 0

This means that when address H'00001003 is set in the break address register (BAR), for example, the bus cycle in which the break condition is satisfied is as follows (where other conditions are met).

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

3. If the data access cycle is selected, the instruction at which the break will occur cannot be determined.

7.4.4 Value of Saved Program Counter

When a break occurs, the address of the instruction from where execution is to be resumed is saved to the stack, and the exception handling state is entered. If the C bus (FAB)/instruction fetch cycle is specified as a break condition, the instruction at which the break should occur can be uniquely determined. If the C bus/data access cycle or I bus/data access cycle is specified as a break condition, the instruction at which the break should occur cannot be uniquely determined.

1. When C bus (FAB)/instruction fetch (before instruction execution) is specified as a break condition:

   The address of the instruction that matched the break condition is saved to the stack. The instruction that matched the condition is not executed, and the break occurs before it. However when a delay slot instruction matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
2. When C bus (FAB)/instruction fetch (after instruction execution) is specified as a break condition:

   The address of the instruction following the instruction that matched the break condition is saved to the stack. The instruction that matches the condition is executed, and the break occurs before the next instruction is executed. However when a delayed branch instruction or delay slot matches the condition, the instruction is executed, and the branch destination address is saved to the stack.
3. When C bus/data access cycle or I bus/data access cycle is specified as a break condition:

   The address after executing several instructions of the instruction that matched the break condition is saved to the stack.

7.4.5 Usage Examples

(1) Break Condition Specified for C Bus Instruction Fetch Cycle

(Example 1-1)

- Register specifications

  BAR_0 = H'00000404, BAMR_0 = H'00000000, BBR_0 = H'0054, BAR_1 = H'00008010, BAMR_1 = H'00000006, BBR_1 = H'0054, BRCR = H'00000020

  <Channel 0>

  Address: H'00000404, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00008010, Address mask: H'00000006

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction of address H'00000404 is executed or before instructions of addresses H'00008010 to H'00008016 are executed.

(Example 1-2)

- Register specifications

  BAR_0 = H'00027128, BAMR_0 = H'00000000, BBR_0 = H'005A, BAR_1= H'00031415, BAMR_1 = H'00000000, BBR_1 = H'0054, BRCR = H'00000000

  <Channel 0>

  Address: H'00027128, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/write/word

  <Channel 1>

  Address: H'00031415, Address mask: H'00000000

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  On channel 0, a user break does not occur since instruction fetch is not a write cycle. On channel 1, a user break does not occur since instruction fetch is performed for an even address.

(Example 1-3)

- Register specifications

  BBR_0 = H'0054, BAR_0 = H'00008404, BAMR_0 = H'00000FFF, BBR_1 = H'0054,
  BAR_1 = H'00008010, BAMR_1 = H'00000006, BRCR = H'00000020

  <Channel 0>

  Address: H'00008404, Address mask: H'00000FFF

  Bus cycle: C bus/instruction fetch (after instruction execution)/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'00008010, Address mask: H'00000006

  Bus cycle: C bus/instruction fetch (before instruction execution)/read (operand size is not included in the condition)

  A user break occurs after an instruction with addresses H'00008000 to H'00008FFE is executed or before an instruction with addresses H'00008010 to H'00008016 are executed.

(2) Break Condition Specified for C Bus Data Access Cycle

(Example 2-1)

- Register specifications

  BBR_0 = H'0064, BAR_0 = H'00123456, BAMR_0 = H'00000000,
  BBR_1 = H'006A, BAR_1 = H'000ABCDE, BAMR_1 = H'000000FF, BRCR = H'00000000

  <Channel 0>

  Address: H'00123456, Address mask: H'00000000

  Bus cycle: C bus/data access/read (operand size is not included in the condition)

  <Channel 1>

  Address: H'000ABCDE, Address mask: H'000000FF

  Bus cycle: C bus/data access/write/word

  On channel 0, a user break occurs with longword read from address H'00123456, word read from address H'00123456, or byte read from address H'00123456. On channel 1, a user break occurs when word is written in addresses H'000ABC00 to H'000ABCFE.

(3) Break Condition Specified for I Bus Data Access Cycle

(Example 3-1)

- Register specifications
  BBR_0 = H'0094, BAR_0 = H'00314156, BAMR_0 = H'00000000,
  BBR_1 = H'12A9, BAR_1 = H'00055555, BAMR_1 = H'00000000, BRCR = H'00000000
  <Channel 0>
  Address: H'00314156, Address mask: H'00000000
  Bus cycle: I bus/instruction fetch/read (operand size is not included in the condition)
  <Channel 1>
  Address: H'00055555, Address mask: H'00000000
  Bus cycle: I bus/data access/write/byte
  On channel 0, the setting of I bus/instruction fetch is ignored.
  On channel 1, a user break occurs when the DMAC writes byte data in address H'00055555 on the I bus (write by the CPU does not generate a user break).

7.5 Interrupt Source

The UBC has the user break source as an interrupt source.

Table 7.4 gives details on this interrupt source.

A user break interrupt is generated when one of the compare match flags (SCMFD3 to SCMFD0 and SCMFC3 to SCMFC0) in the break control register (BRCR) is set to 1. Clearing the interrupt flag bit to 0 cancels the interrupt request.

Table 7.4 Interrupt Source

Abbreviation	Interrupt Source	Interrupt Enable Bit	Interrupt Flag	Interrupt Level
User break	User break interrupt	—	SCMFD3, SCMFD2, SCMFD1, SCMFD0, SCMFC3, SCMFC2, SCMFC1, SCMFC0	Fixed to 15

7.6 Usage Notes

1. The CPU can read from or write to the UBC registers via the I bus. Accordingly, during the period from executing an instruction to rewrite the UBC register till the new value is actually rewritten, the desired break may not occur. In order to know the timing when the UBC register is changed, read from the last written register. Instructions after then are valid for the newly written register value.
2. The UBC cannot monitor access to the C bus and I bus cycles in the same channel.
3. When a user break and another exception occur at the same instruction, which has higher priority is determined according to the priority levels defined in table 5.1 in section 5, Exception Handling. If an exception with a higher priority occurs, the user break does not occur.
4. Note the following when a break occurs in a delay slot.

   If a pre-execution break is set at a delay slot instruction, the break is not generated until immediately before execution of the branch destination.
5. User breaks are disabled during UBC module standby mode. Do not read from or write to the UBC registers during UBC module standby mode; the values are not guaranteed.
6. Do not set an address within an interrupt exception handling routine whose interrupt priority level is at least 15 (including user break interrupts) as a break address.
7. Do not set break after instruction execution for the SLEEP instruction or for the delayed branch instruction where the SLEEP instruction is placed at its delay slot.
8. When setting a break for a 32-bit instruction, set the address where the upper 16 bits are placed. If the address of the lower 16 bits is set and a break before instruction execution is set as a break condition, the break is handled as a break after instruction execution.
9. Do not set a pre-execution break for an instruction that immediately follows a DIVU or DIVS instruction. If such a break is set and an interrupt or other exception occurs during execution of the DIVU or DIVS instruction, the pre-execution break will still occur even though execution of the DIVU or DIVS instruction is suspended.
10. Do not set a pre- and post-execution break for the same address at the same time. For example, if a pre-execution break for channel 0 and a post -execution break for channel 1 are set for the same address at the same time, the condition match flags on channel 1 after instruction execution will be set even though a pre-execution break has occurred on channel 0.

Section 8 Data Transfer Controller (DTC)

This LSI includes a data transfer controller (DTC). The DTC can be activated to transfer data by an interrupt request.

8.1 Features

- Transfer possible over any number of channels
- Chain transfer

  Multiple rounds of data transfer is executed in response to a single activation source

  Chain transfer is only possible after data transfer has been done for the specified number of times (i.e. when the transfer counter is 0)
- Three transfer modes

  Normal/repeat/block transfer modes selectable

  Transfer source and destination addresses can be selected from increment/decrement/fixed
- The transfer source and destination addresses can be specified by 32 bits to select a 4-Gbyte address space directly
- Size of data for data transfer can be specified as byte, word, or longword
- A CPU interrupt can be requested for the interrupt that activated the DTC

  A CPU interrupt can be requested after one data transfer completion

  A CPU interrupt can be requested after the specified data transfer completion
- Read skip of the transfer information specifiable
- Write-back skip executed for the fixed transfer source and destination addresses
- Module stop mode specifiable
- Short address mode specifiable
- Bus release timing selectable: Three types
- DTC activation priority selectable: Two types

Figure 8.1 shows a block diagram of the DTC. The DTC transfer information can be allocated to the data area*.

Note: When the transfer information is stored in the on-chip RAM, the RAME bits in SYSCR1 and SYSCR2 must be set to 1.

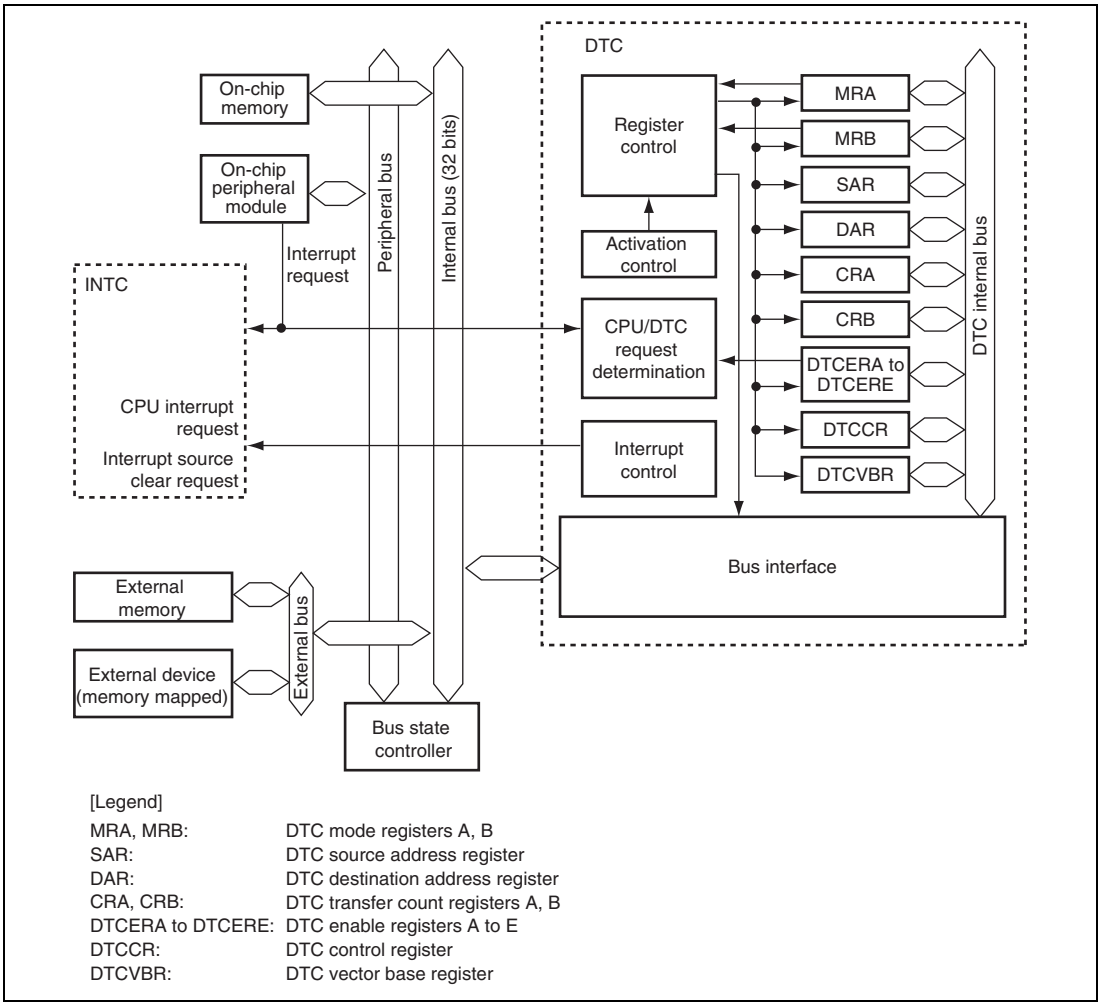


Figure 8.1 Block Diagram of DTC

8.2 Register Descriptions

DTC has the following registers. For details on the addresses of these registers and the states of these registers in each processing state, see section 32, List of Registers.

These six registers MRA, MRB, SAR, DAR, CRA, and CRB cannot be directly accessed by the CPU. The contents of these registers are stored in the data area as transfer information. When a DTC activation request occurs, the DTC reads a start address of transfer information that is stored in the data area according to the vector address, reads the transfer information, and transfers data. After the data transfer is complete, it writes a set of updated transfer information back to the data area.

On the other hand, DTCERA to DTCERE, DTCCR, and DTCVBR can be directly accessed by the CPU.

Table 8.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
DTC enable register A	DTCERA	R/W	H'0000	H'FFFE6000	8, 16
DTC enable register B	DTCERB	R/W	H'0000	H'FFFE6002	8, 16
DTC enable register C	DTCERC	R/W	H'0000	H'FFFE6004	8, 16
DTC enable register D	DTCERD	R/W	H'0000	H'FFFE6006	8, 16
DTC enable register E	DTCERE	R/W	H'0000	H'FFFE6008	8, 16
DTC control register	DTCCR	R/W	H'00	H'FFFE6010	8
DTC vector base register	DTCVBR	R/W	H'00000000	H'FFFE6014	8, 16, 32
Bus function extending register	BSCEHR	R/W	H'0000	H'FFFE3C1A	16

8.2.1 DTC Mode Register A (MRA)

MRA selects DTC operating mode. MRA cannot be accessed directly by the CPU.

Bit:	7	6	5	4	3	2	1	0
	MD[1:0]		Sz[1:0]		SM[1:0]		-	-
Initial value:	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-

*: Undefined

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MD[1:0]	Undefined	—	DTC Mode 1 and 0 Specify DTC transfer mode. 00: Normal mode 01: Repeat mode 10: Block transfer mode 11: Setting prohibited
5, 4	Sz[1:0]	Undefined	—	DTC Data Transfer Size 1 and 0 Specify the size of data to be transferred. 00: Byte-size transfer 01: Word-size transfer 10: Longword-size transfer 11: Setting prohibited
3, 2	SM[1:0]	Undefined	—	Source Address Mode 1 and 0 Specify an SAR operation after a data transfer. 0x: SAR is fixed (SAR write-back is skipped) 10: SAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
x: Don't care

8.2.2 DTC Mode Register B (MRB)

MRB selects DTC operating mode. MRB cannot be accessed directly by the CPU.

Bit:	7	6	5	4	3	2	1	0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		-	-
Initial value:	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-

*: Undefined

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	DTC Chain Transfer Enable Specifies the chain transfer. For details, see section 8.5.6, Chain Transfer. The chain transfer condition is selected by the CHNS bit. 0: Disables the chain transfer 1: Enables the chain transfer
6	CHNS	Undefined	—	DTC Chain Transfer Select Specifies the chain transfer condition. If the following transfer is a chain transfer, the completion check of the specified transfer count is not performed and activation source flag or DTCER is not cleared. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0
5	DISEL	Undefined	—	DTC Interrupt Select When this bit is set to 1, an interrupt request is generated to the CPU every time a data transfer or a block data transfer ends. When this bit is set to 0, a CPU interrupt request is only generated when the specified number of data transfers ends.

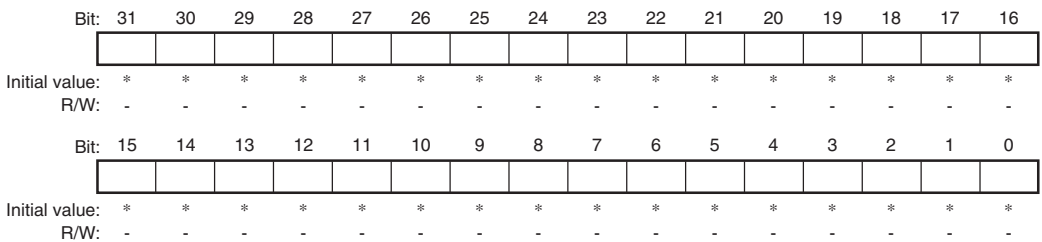
Bit	Bit Name	Initial Value	R/W	Description
4	DTS	Undefined	—	DTC Transfer Mode Select Specifies either the source or destination as repeat or block area during repeat or block transfer mode. 0: Specifies the destination as repeat or block area 1: Specifies the source as repeat or block area
3, 2	DM[1:0]	Undefined	—	Destination Address Mode 1 and 0 Specify a DAR operation after a data transfer. 0x: DAR is fixed (DAR write-back is skipped) 10: DAR is incremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10) 11: SAR is decremented after a transfer (by 1 when Sz1 and Sz0 = B'00; by 2 when Sz1 and Sz0 = B'01; by 4 when Sz1 and Sz0 = B'10)
1, 0	—	Undefined	—	Reserved The write value should always be 0.

[Legend]
x: Don't care

8.2.3 DTC Source Address Register (SAR)

SAR is a 32-bit register that designates the source address of data to be transferred by the DTC.

SAR cannot be accessed directly from the CPU.



*: Undefined

8.2.4 DTC Destination Address Register (DAR)

DAR is a 32-bit register that designates the destination address of data to be transferred by the DTC.

DAR cannot be accessed directly from the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*: Undefined

8.2.5 DTC Transfer Count Register A (CRA)

CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal transfer mode, CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and bit DTCEn (n = 15 to 0) corresponding to the activation source is cleared and then an interrupt is requested to the CPU when the count reaches H'0000. The transfer count is 1 when CRA = H'0001, 65,535 when CRA = H'FFFF, and 65,536 when CRA = H'0000.

In repeat transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and the lower eight bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent to CRAL when the count reaches H'00. The transfer count is 1 when CRAH = CRAL = H'01, 255 when CRAH = CRAL = H'FF, and 256 when CRAH = CRAL = H'00.

In block transfer mode, CRA is divided into two parts: the upper eight bits (CRAH) and the lower eight bits (CRAL). CRAH holds the block size while CRAL functions as an 8-bit block-size counter (1 to 256 for byte, word, or longword). CRAL is decremented by 1 every time a byte (word or longword) data is transferred, and the contents of CRAH are sent to CRAL when the count reaches H'00. The block size is 1 byte (word or longword) when CRAH = CRAL =H'01, 255 bytes (words or longwords) when CRAH = CRAL = H'FF, and 256 bytes (words or longwords) when CRAH = CRAL =H'00.

CRA cannot be accessed directly from the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*: Undefined

8.2.6 DTC Transfer Count Register B (CRB)

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time a block of data is transferred, and bit DTCEn (n = 15 to 0) corresponding to the activation source is cleared and then an interrupt is requested to the CPU when the count reaches H'0000. The transfer count is 1 when CRB = H'0001, 65,535 when CRB = H'FFFF, and 65,536 when CRB = H'0000.

CRB is not available in normal and repeat modes and cannot be accessed directly by the CPU.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

*: Undefined

8.2.7 DTC Enable Registers A to E (DTCERA to DTCERE)

DTCER which is comprised of eight registers, DTCERA to DTCERE, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 8.2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	DTCE15	0	R/W	DTC Activation Enable 15 to 0 Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source. [Clearing conditions] • When writing 0 to the bit to be cleared after reading 1 • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended [Setting condition] • Writing 1 to the bit after reading 0
14	DTCE14	0	R/W	
13	DTCE13	0	R/W	
12	DTCE12	0	R/W	
11	DTCE11	0	R/W	
10	DTCE10	0	R/W	
9	DTCE9	0	R/W	
8	DTCE8	0	R/W	
7	DTCE7	0	R/W	
6	DTCE6	0	R/W	
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

8.2.8 DTC Control Register (DTCCR)

DTCCR specifies transfer information read skip.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	RRS	RCHNE	-	-	ERR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R	R/(W)*

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRS	0	R/W	DTC Transfer Information Read Skip Enable Controls the vector address read and transfer information read. A DTC vector number is always compared with the vector number for the previous activation. If the vector numbers match and this bit is set to 1, the DTC data transfer is started without reading a vector address and transfer information. If the previous DTC activation is a chain transfer, the vector address read and transfer information read are always performed. 0: Transfer read skip is not performed. 1: Transfer read skip is performed when the vector numbers match.
3	RCHNE	0	R/W	Chain Transfer Enable After DTC Repeat Transfer Enables/disables the chain transfer while transfer counter (CRAL) is 0 in repeat transfer mode. In repeat transfer mode, the CRAH value is written to CRAL when CRAL is 0. Accordingly, chain transfer may not occur when CRAL is 0. If this bit is set to 1, the chain transfer is enabled when CRAH is written to CRAL. 0: Disables the chain transfer after repeat transfer 1: Enables the chain transfer after repeat transfer
2, 1	—	All 0	R	Reserved These are read-only bits and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
0	ERR	0	R/(W)*	Transfer Stop Flag Indicates that a DTC address error or NMI interrupt has occurred. If a DTC address error or NMI interrupt occurs while the DTC is active, a DTC address error handling or NMI interrupt handling processing is executed after the DTC has released the bus mastership. The DTC halts after a data transfer or a transfer information writing state depending on the NMI input timing. Note that a writing state is not exact, when the DTC halts after a data transfer. When the data is transferred, set a transfer information once again (except that a read skip is performed). 0: No interrupt has occurred 1: An interrupt has occurred [Clearing condition] • When writing 0 after reading 1

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

8.2.9 DTC Vector Base Register (DTCVBR)

DTCVBR is a 32-bit register that specifies the base address for vector table address calculation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12		All 0	R/W	Bits 11 to 0 are always read as 0. The write value should always be 0.
11 to 0	—	All 0	R	

8.2.10 Bus Function Extending Register (BSCEHR)

BSCEHR is a 16-bit register that specifies the timing of bus release by the DTC and other functions. This register should be used to give priority to the DTC transfer or reduce the number of cycles in which the DTC is active. For more details, see section 9.4.8, Bus Function Extending Register (BSCEHR).

8.3 Activation Sources

The DTC is activated by an interrupt request. The interrupt source is selected by DTCER. A DTC activation source can be selected by setting the corresponding bit in DTCER; the CPU interrupt source can be selected by clearing the corresponding bit in DTCER. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source interrupt flag or corresponding DTCER bit is cleared.

8.4 Location of Transfer Information and DTC Vector Table

Locate the transfer information in the data area. The start address of transfer information should be located at the address that is a multiple of four (4n). Otherwise, the lower two bits are ignored during access ([1:0] = B'00.) Transfer information located in the data area is shown in figure 8.2.

Short address mode can be selected by setting the DTSA bit in the bus function extending register (BSCEHR) to 1 only when all DTC transfer sources and destinations are located in the on-chip RAM and on-chip peripheral module areas (see section 9.4.8, Bus Function Extending Register (BSCEHR)).

In normal transfer, four longwords should be read as the transfer information; in short address mode, the transfer information is reduced to three longwords and the DTC active period becomes shorter.

The DTC reads the start address of transfer information from the vector table according to the activation source, and then reads the transfer information from the start address. Figure 8.3 shows correspondences between the DTC vector address and transfer information.

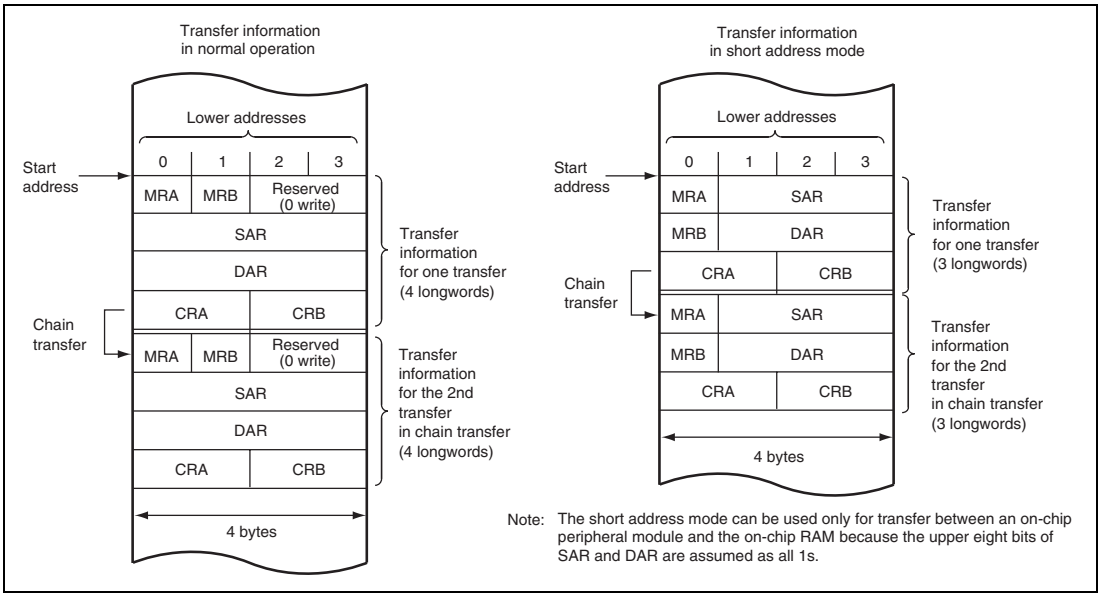


Figure 8.2 Transfer Information on Data Area

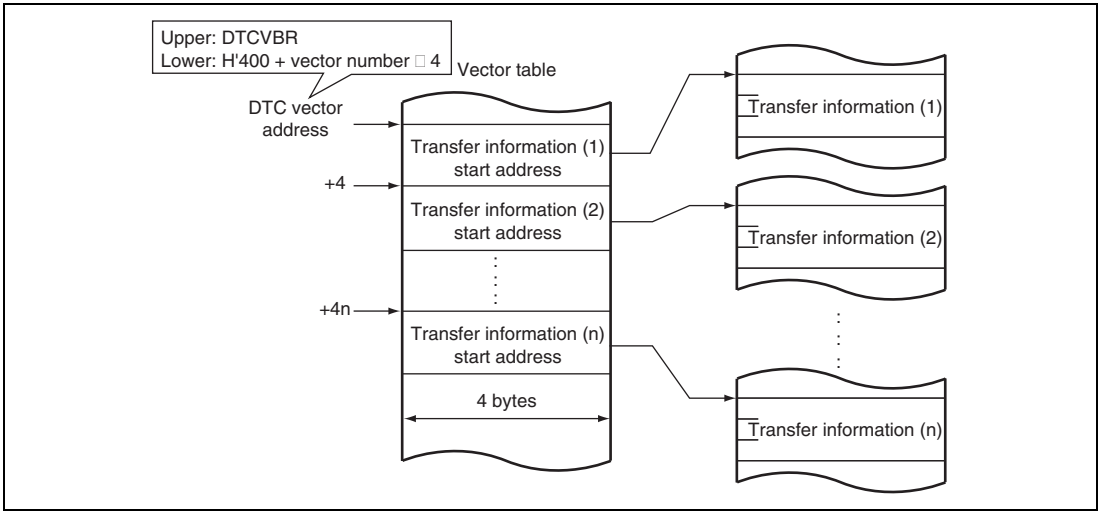


Figure 8.3 Correspondence between DTC Vector Address and Transfer Information

Table 8.2 shows correspondence between the DTC activation source and vector address.

Table 8.2 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE*1	Transfer Source	Transfer Destination	Priority
External pin	IRQ0	64	H'00000500	DTCERA15	Any location*2	Any location*2	High
	IRQ1	65	H'00000504	DTCERA14	Any location*2	Any location*2	
	IRQ2	66	H'00000508	DTCERA13	Any location*2	Any location*2	
	IRQ3	67	H'0000050C	DTCERA12	Any location*2	Any location*2	
	IRQ4	68	H'00000510	DTCERA11	Any location*2	Any location*2	
	IRQ5	69	H'00000514	DTCERA10	Any location*2	Any location*2	
	IRQ6	70	H'00000518	DTCERA9	Any location*2	Any location*2	
	IRQ7	71	H'0000051C	DTCERA8	Any location*2	Any location*2	
A/D	ADI0	92	H'00000570	DTCERA7	ADDR0 to ADDR3	Any location*2	
	ADI1	96	H'00000580	DTCERA6	ADDR4 to ADDR7	Any location*2	
RCAN-ET	RM0_0	106	H'000005A8	DTCERA4	CONTROL0H to CONTROL1L*3	Any location*2	
CMT	CMI0	140	H'00000630	DTCERA3	Any location*2	Any location*2	
	CMI1	144	H'00000640	DTCERA2	Any location*2	Any location*2	
USB	USBRXI1	150	H'00000658	DTCERE7	USBEPDR4	Any location*2	
	USBTXI1	151	H'0000065C	DTCERE6	Any location*2	USBEPDR5	
	USBRXI0	154	H'00000668	DTCERA1	USBEPDR1	Any location*2	
	USBTXI0	155	H'0000066C	DTCERA0	Any location*2	USBEPDR2	
MTU2_CH0	TGIA_0	156	H'00000670	DTCERB15	Any location*2	Any location*2	
	TGIB_0	157	H'00000674	DTCERB14	Any location*2	Any location*2	
	TGIC_0	158	H'00000678	DTCERB13	Any location*2	Any location*2	
	TGID_0	159	H'0000067C	DTCERB12	Any location*2	Any location*2	
MTU2_CH 1	TGIA_1	164	H'00000690	DTCERB11	Any location*2	Any location*2	
	TGIB_1	165	H'00000694	DTCERB10	Any location*2	Any location*2	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE*1	Transfer Source	Transfer Destination	Priority
MTU2_CH2	TGIA_2	172	H'000006B0	DTCERB9	Any location*2	Any location*2	High
	TGIB_2	173	H'000006B4	DTCERB8	Any location*2	Any location*2	
MTU2_CH3	TGIA_3	180	H'000006D0	DTCERB7	Any location*2	Any location*2	
	TGIB_3	181	H'000006D4	DTCERB6	Any location*2	Any location*2	
	TGIC_3	182	H'000006D8	DTCERB5	Any location*2	Any location*2	
	TGID_3	183	H'000006DC	DTCERB4	Any location*2	Any location*2	
MTU2_CH4	TGIA_4	188	H'000006F0	DTCERB3	Any location*2	Any location*2	
	TGIB_4	189	H'000006F4	DTCERB2	Any location*2	Any location*2	
	TGIC_4	190	H'000006F8	DTCERB1	Any location*2	Any location*2	
	TGID_4	191	H'000006FC	DTCERB0	Any location*2	Any location*2	
	TCIV_4	192	H'00000700	DTCERC15	Any location*2	Any location*2	
MTU2_CH5	TGIU_5	196	H'00000710	DTCERC14	Any location*2	Any location*2	
	TGIV_5	197	H'00000714	DTCERC13	Any location*2	Any location*2	
	TGIW_5	198	H'00000718	DTCERC12	Any location*2	Any location*2	
MTU2S_CH3	TGIA_3S	204	H'00000730	DTCERC3	Any location*2	Any location*2	
	TGIB_3S	205	H'00000734	DTCERC2	Any location*2	Any location*2	
	TGIC_3S	206	H'00000738	DTCERC1	Any location*2	Any location*2	
	TGID_3S	207	H'0000073C	DTCERC0	Any location*2	Any location*2	
MTU2S_CH4	TGIA_4S	212	H'00000750	DTCERD15	Any location*2	Any location*2	
	TGIB_4S	213	H'00000754	DTCERD14	Any location*2	Any location*2	
	TGIC_4S	214	H'00000758	DTCERD13	Any location*2	Any location*2	
	TGID_4S	215	H'0000075C	DTCERD12	Any location*2	Any location*2	
	TCIV_4S	216	H'00000760	DTCERD11	Any location*2	Any location*2	
MTU2S_CH5	TGIU_5S	220	H'00000770	DTCERD10	Any location*2	Any location*2	
	TGIV_5S	221	H'00000774	DTCERD9	Any location*2	Any location*2	
	TGIW_5S	222	H'00000778	DTCERD8	Any location*2	Any location*2	
IIC3	RXI	230	H'00000798	DTCERD7	ICDRR	Any location*2	
	TXI	231	H'0000079C	DTCERD6	Any location*2	ICDRT	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE*1	Transfer Source	Transfer Destination	Priority
RSPI	SPRI	234	H'000007A8	DTCERD5	SPDR	Any location*2	High
	SPTI	235	H'000007AC	DTCERD4	Any location*2	SPDR	
SCI4	RXI4	237	H'000007B4	DTCERD3	SCRDR_4	Any location*2	
	TXI4	238	H'000007B8	DTCERD2	Any location*2	SCTDR_4	
SCI0	RXI0	241	H'000007C4	DTCERE15	SCRDR_0	Any location*2	
	TXI0	242	H'000007C8	DTCERE14	Any location*2	SCTDR_0	
SCI1	RXI1	245	H'000007D4	DTCERE13	SCRDR_1	Any location*2	
	TXI1	246	H'000007D8	DTCERE12	Any location*2	SCTDR_1	
SCI2	RXI2	249	H'000007E4	DTCERE11	SCRDR_2	Any location*2	
	TXI2	250	H'000007E8	DTCERE10	Any location*2	SCTDR_2	
SCIF3	RXI3	254	H'000007F8	DTCERE9	SCFRDR_3	Any location*2	
	TXI3	255	H'000007FC	DTCERE8	Any location*2	SCFTDR_3	Low

Notes: 1. The DTCE bits with no corresponding interrupt are reserved, and the write value should always be 0.

2. An external memory, a memory-mapped external device, an on-chip memory, or an on-chip peripheral module (except for DTC, BSC, UBC, AUD, FLASH, and DMAC) can be selected as the source or destination. Note that at least either the source or destination must be an on-chip peripheral module; transfer cannot be done among an external memory, a memory-mapped external device, and an on-chip memory.

3. Read to a message control field in mailbox 0 by using a block transfer mode or etc.

8.5 Operation

There are three transfer modes: normal, repeat, and block. Since transfer information is in the data area, it is possible to transfer data over any required number of channels. When activated, the DTC reads the transfer information stored in the data area and transfers data according to the transfer information. After the data transfer is complete, it writes updated transfer information back to the data area.

The DTC specifies the source address and destination address in SAR and DAR, respectively. After a transfer, SAR and DAR are incremented, decremented, or fixed independently.

Table 8.3 shows the DTC transfer modes.

Table 8.3 DTC Transfer Modes

Transfer Mode	Size of Data Transferred at One Transfer Request	Memory Address Increment or Decrement	Transfer Count
Normal	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1 to 65536
Repeat*1	1 byte/word/longword	Incremented/decremented by 1, 2, or 4, or fixed	1 to 256*3
Block*2	Block size specified by CRAH (1 to 256 bytes/words/longwords)	Incremented/decremented by 1, 2, or 4, or fixed	1 to 65536*4

Notes: 1. Either source or destination is specified to repeat area.
2. Either source or destination is specified to block area.
3. After transfer of the specified transfer count, initial state is recovered to continue the operation.
4. Number of transfers of the specified block size of data

Setting the CHNE bit in MRB to 1 makes it possible to perform a number of transfers with a single activation (chain transfer). Setting the CHNS bit in MRB to 1 can also be made to have chain transfer performed only when the transfer counter value is 0.

Figure 8.4 shows a flowchart of DTC operation, and table 8.4 summarizes the conditions for DTC transfers including chain transfer (combinations for performing the second and third transfers are omitted).

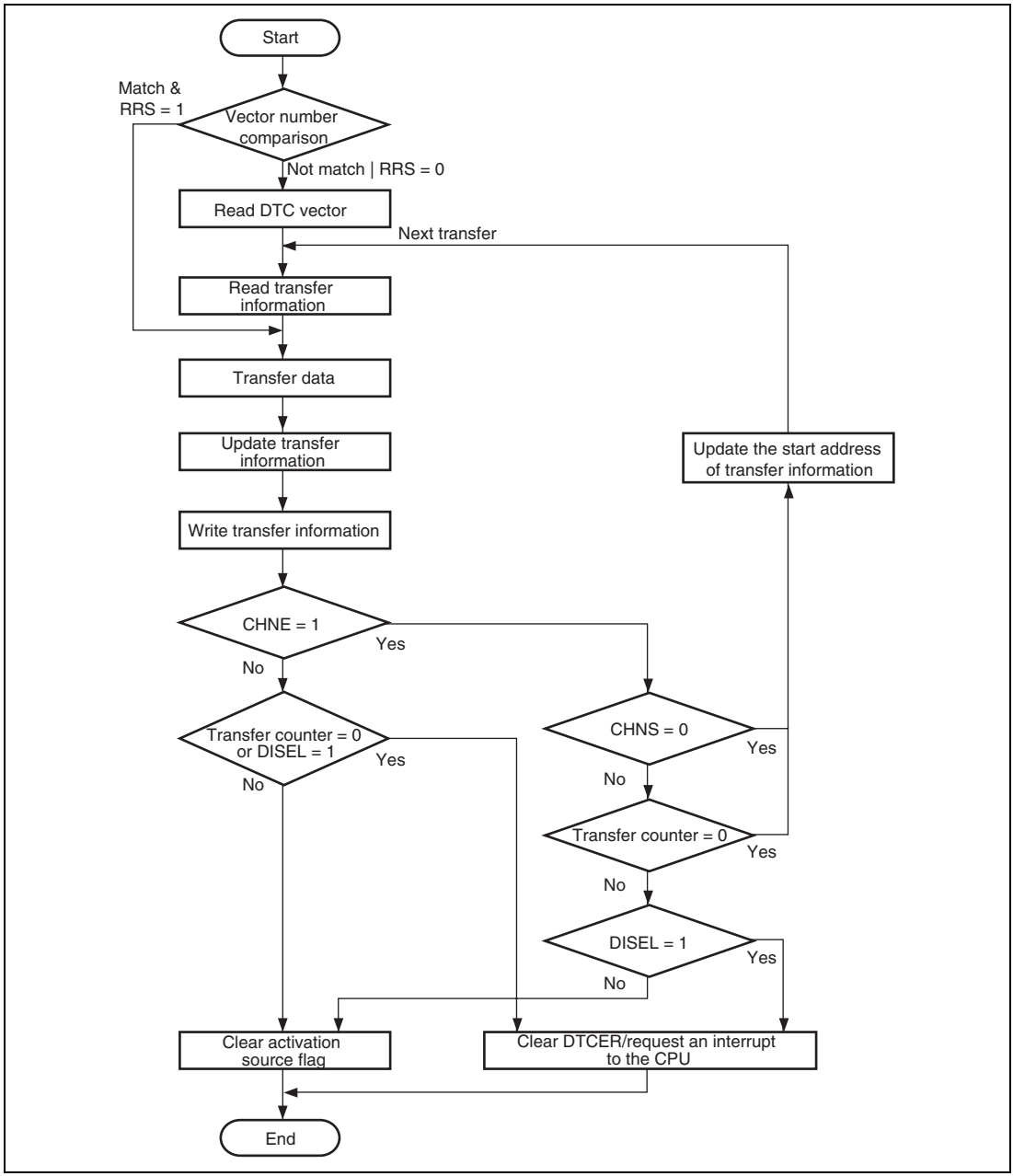


Figure 8.4 Flowchart of DTC Operation

Table 8.4 DTC Transfer Conditions (Chain Transfer Conditions Included)

	1st Transfer					2nd Transfer					
Transfer Mode	CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	DTC Transfer
Normal	0	—	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	0	—	—	0	0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	
	1	1	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	1	1	—	1	Not 0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	—	—	0	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	

	1st Transfer					2nd Transfer					
Transfer Mode	**CHNE**	**CHNS**	**RCHNE**	**DISEL**	**Transfer Counter**[*1]	**CHNE**	**CHNS**	**RCHNE**	**DISEL**	**Transfer Counter**[*1]	**DTC Transfer**
Repeat	0	—	—	0	—	—	—	—	—	—	Ends at 1st transfer
	0	—	—	1	—	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	0	—	—	—	0	—	—	0	—	Ends at 2nd transfer
						0	—	—	1	—	Ends at 2nd transfer Interrupt request to CPU
	1	1	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	1	1	—	1	Not 0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	0	0	0[*2]	—	—	—	—	—	Ends at 1st transfer
	1	1	0	1	0[*2]	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	1	—	0[*2]	0	—	—	0	—	Ends at 2nd transfer
						0	—	—	1	—	Ends at 2nd transfer Interrupt request to CPU

Transfer Mode	1st Transfer CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	2nd Transfer CHNE	CHNS	RCHNE	DISEL	Transfer Counter*1	DTC Transfer
Block	0	—	—	0	Not 0	—	—	—	—	—	Ends at 1st transfer
	0	—	—	0	0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	
	1	1	—	0	—	—	—	—	—	—	Ends at 1st transfer
	1	1	—	1	Not 0	—	—	—	—	—	Ends at 1st transfer Interrupt request to CPU
	1	1	—	1	0	0	—	—	0	Not 0	Ends at 2nd transfer
						0	—	—	0	0	Ends at 2nd transfer Interrupt request to CPU
						0	—	—	1	—	

Notes: 1. CRA in normal mode transfer, CRAL in repeat transfer mode, or CRB in block transfer mode

2. When the contents of the CRAH is written to the CRAL

8.5.1 Transfer Information Read Skip Function

By setting the RRS bit of DTCCR, the vector address read and transfer information read can be skipped. The current DTC vector number is always compared with the vector number of previous activation. If the vector numbers match when RRS = 1, a DTC data transfer is performed without reading the vector address and transfer information. If the previous activation is a chain transfer, the vector address read and transfer information read are always performed. Figure 8.5 shows the transfer information read skip timing.

To modify the vector table and transfer information, temporarily clear the RRS bit to 0, modify the vector table and transfer information, and then set the RRS bit to 1 again. When the RRS bit is cleared to 0, the stored vector number is deleted, and the updated vector table and transfer information are read at the next activation.

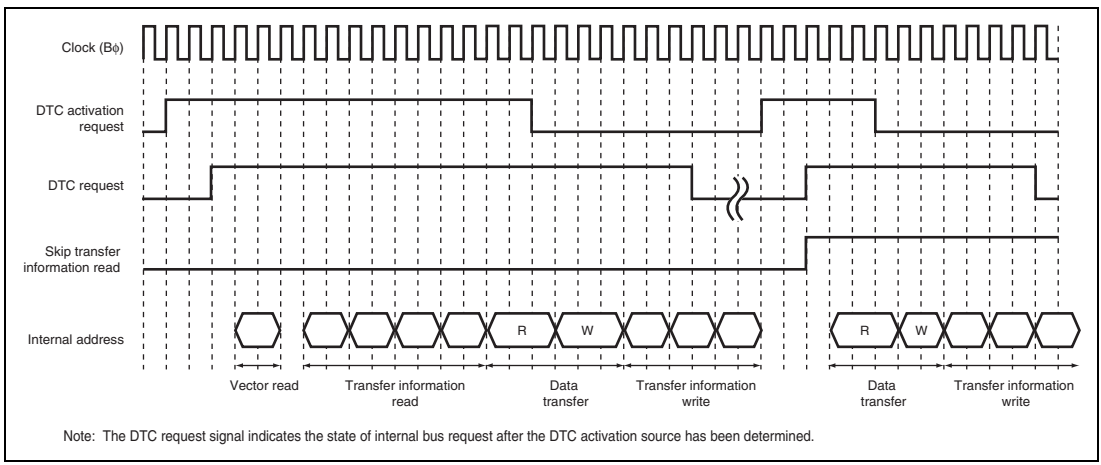


Figure 8.5 Transfer Information Read Skip Timing (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)

8.5.2 Transfer Information Write-Back Skip Function

By specifying bit SM1 in MRA and bit DM1 in MRB to the fixed address mode, a part of transfer information will not be written back. Table 8.5 shows the transfer information write-back skip condition and write-back skipped registers. Note that the CRA and CRB are always written back. The write-back of the MRA and MRB are always skipped.

Table 8.5 Transfer Information Write-Back Skip Condition and Write-Back Skipped Registers

SM1	DM1	SAR	DAR
0	0	Skipped	Skipped
0	1	Skipped	Written back
1	0	Written back	Skipped
1	1	Written back	Written back

8.5.3 Normal Transfer Mode

In normal transfer mode, data are transferred in one byte, one word, or one longword units in response to a single activation request. From 1 to 65,536 transfers can be specified. The transfer source and destination addresses can be specified as incremented, decremented, or fixed. When the specified number of transfers ends, an interrupt can be requested to the CPU.

Table 8.6 lists the register function in normal transfer mode. Figure 8.6 shows the memory map in normal transfer mode.

Table 8.6 Register Function in Normal Transfer Mode

Register	Function	Written Back Value
SAR	Source address	Incremented/decremented/fixed*
DAR	Destination address	Incremented/decremented/fixed*
CRA	Transfer count A	CRA – 1
CRB	Transfer count B	Not updated

Note: * Transfer information write-back is skipped.

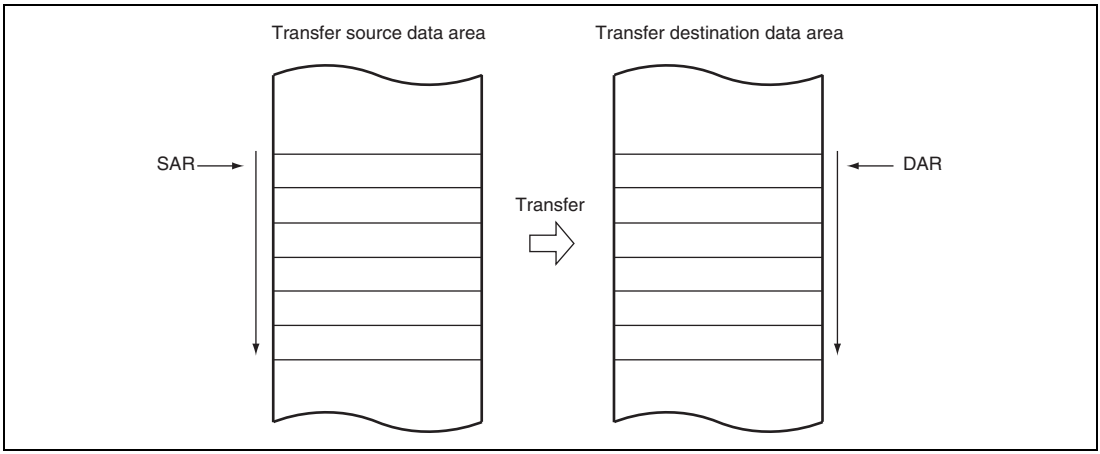


Figure 8.6 Memory Map in Normal Transfer Mode

8.5.4 Repeat Transfer Mode

In repeat transfer mode, data are transferred in one byte, one word, or one longword units in response to a single activation request. By the DTS bit in MRB, either the source or destination can be specified as a repeat area. From 1 to 256 transfers can be specified. When the specified number of transfers ends, the transfer counter and address register specified as the repeat area is restored to the initial state, and transfer is repeated. The other address register is then incremented, decremented, or left fixed. In repeat transfer mode, the transfer counter (CRAL) is updated to the value specified in CRAH when CRAL becomes H'00. Thus the transfer counter value does not reach H'00, and therefore a CPU interrupt cannot be requested when DISEL = 0.

Table 8.7 lists the register function in repeat transfer mode. Figure 8.7 shows the memory map in repeat transfer mode.

Table 8.7 Register Function in Repeat Transfer Mode

		Written Back Value	
Register	**Function**	**CRAL is not 1**	**CRAL is 1**
SAR	Source address	Incremented/decremented/fixed*	DTS = 0: Incremented/ decremented/fixed* DTS = 1: SAR initial value
DAR	Destination address	Incremented/decremented/fixed*	DTS = 0: DAR initial value DTS = 1: Incremented/ decremented/fixed*
CRAH	Transfer count storage	CRAH	CRAH
CRAL	Transfer count A	CRAL – 1	CRAH
CRB	Transfer count B	Not updated	Not updated

Note: * Transfer information write-back is skipped.

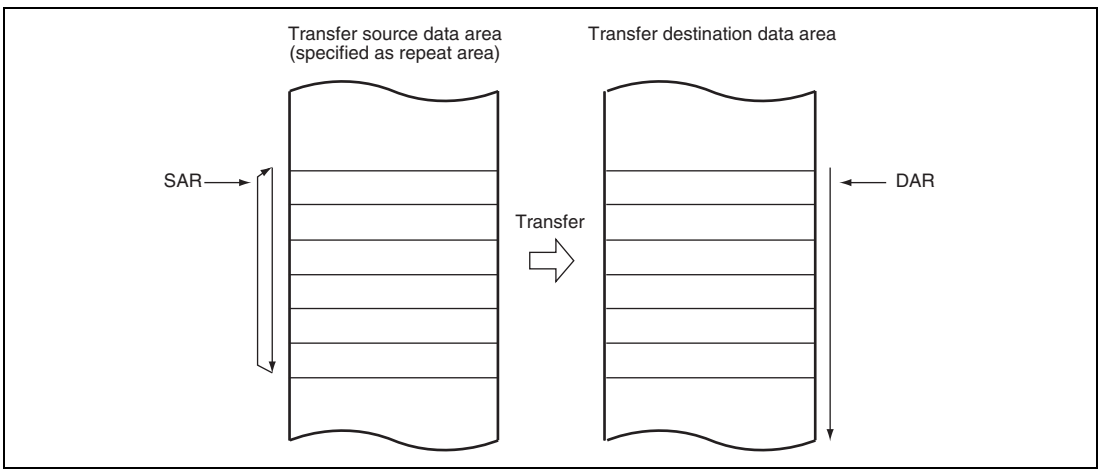


Figure 8.7 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

8.5.5 Block Transfer Mode

In block transfer mode, data are transferred in block units in response to a single activation request. Either the transfer source or the transfer destination is designated as a block area by the DTS bit in MRB.

The block size is 1 to 256 bytes (1 to 256 words, or 1 to 256 longwords). When transfer of one block of data ends, the block size counter (CRAL) and address register (SAR when DTS = 1 or DAR when DTS = 0) for the area specified as the block area are initialized. The other address register is then incremented, decremented, or left fixed. From 1 to 65,536 transfers can be specified. When the specified number of transfers ends, an interrupt is requested to the CPU.

Table 8.8 lists the register function in block transfer mode. Figure 8.8 shows the memory map in block transfer mode.

Table 8.8 Register Function in Block Transfer Mode

Register	Function	Written Back Value
SAR	Source address	DTS = 0: Incremented/decremented/fixed*
		DTS = 1: SAR initial value
DAR	Destination address	DTS = 0: DAR initial value
		DTS = 1: Incremented/decremented/fixed*
CRAH	Block size storage	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB – 1

Note: * Transfer information write-back is skipped.

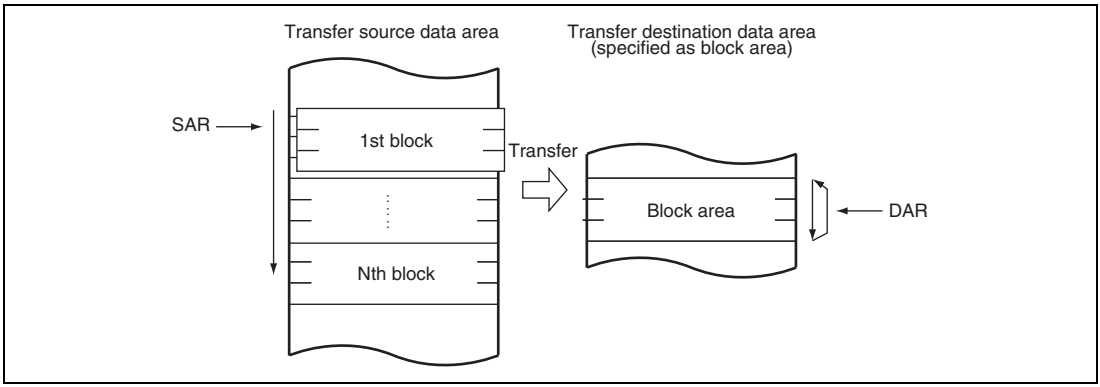


Figure 8.8 Memory Map in Block Transfer Mode (When Transfer Destination is Specified as Block Area)

8.5.6 Chain Transfer

Setting the CHNE bit in MRB to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. Setting the CHNE and CHNS bits in MRB set to 1 enables a chain transfer only when the transfer counter reaches 0. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently. Figure 8.9 shows the chain transfer operation.

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting the DISEL bit to 1, and the interrupt source flag for the activation source and DTCER are not affected.

In repeat transfer mode, setting the RCHNE bit in DTCCR and the CHNE and CHNS bits in MRB to 1 enables a chain transfer after transfer with transfer counter = 1 has been completed.

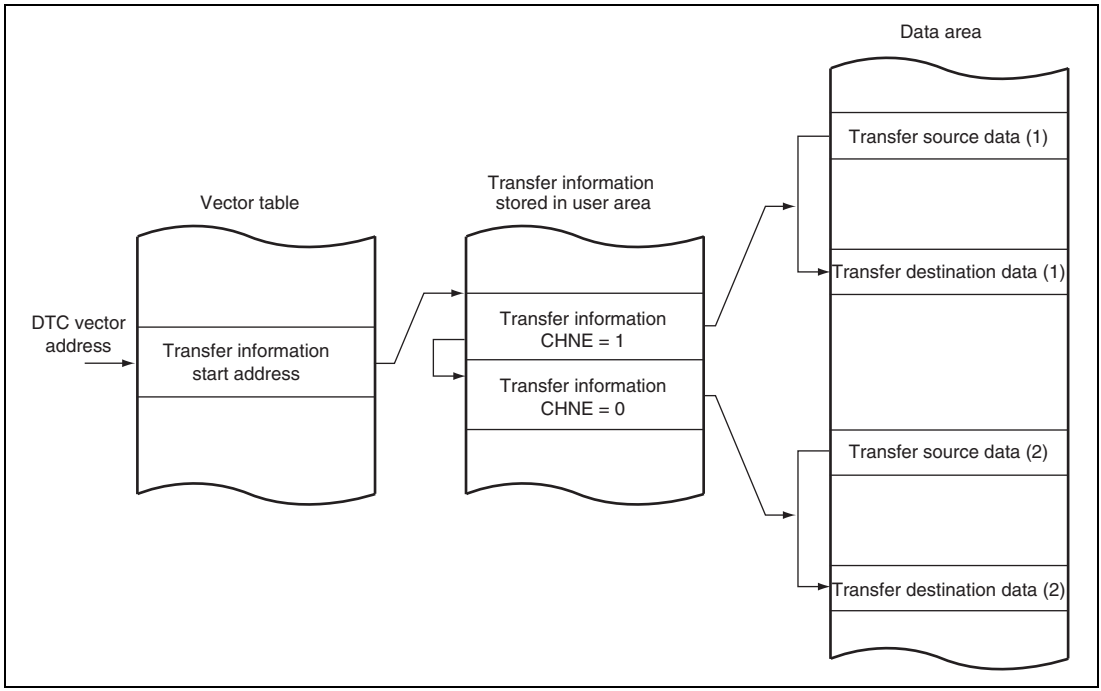
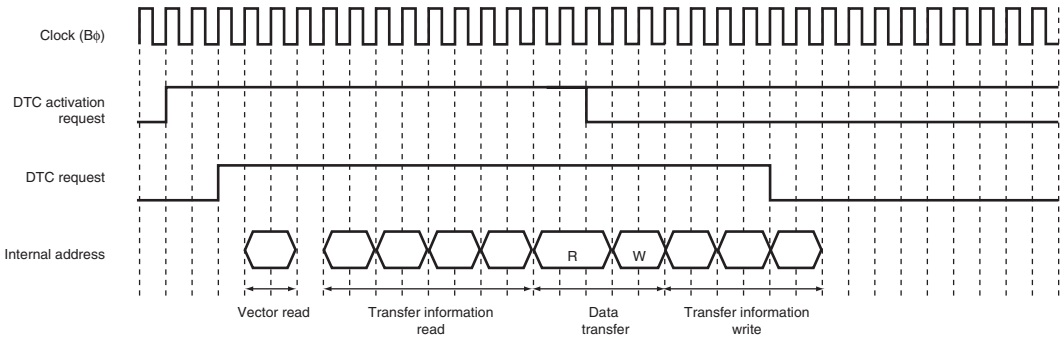


Figure 8.9 Operation of Chain Transfer

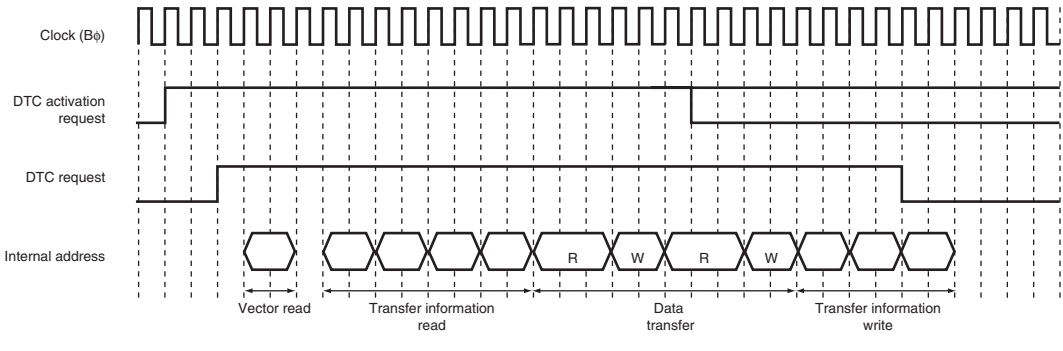
8.5.7 Operation Timing

Figures 8.10 to 8.15 show the DTC operation timings.



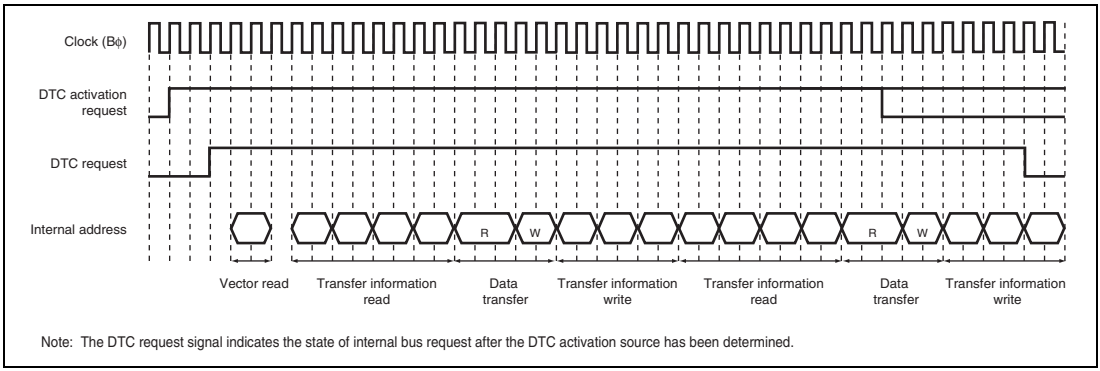
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.10 Example of DTC Operation Timing: Normal Transfer Mode or Repeat Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)



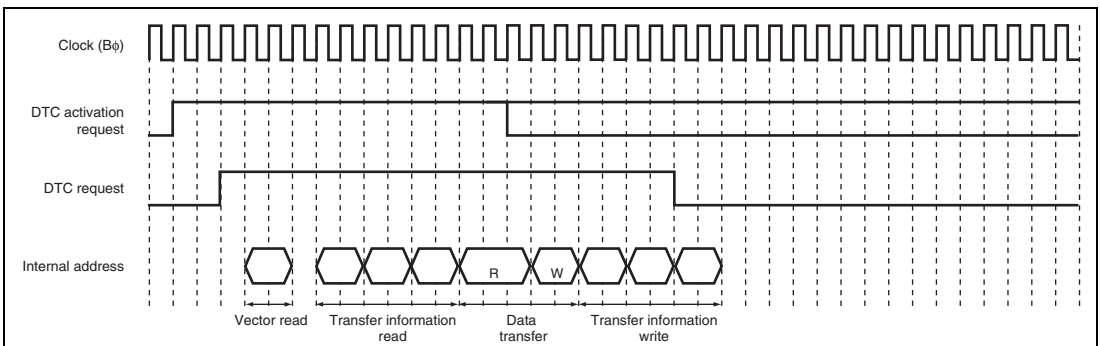
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.11 Example of DTC Operation Timing: Block Transfer Mode with Block Size = 2 (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)



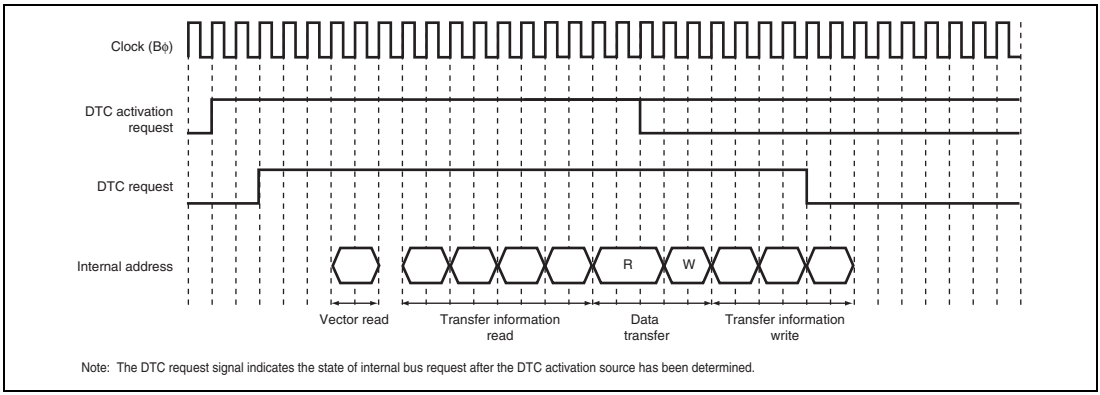
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.12 Example of DTC Operation Timing: Chain Transfer (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)



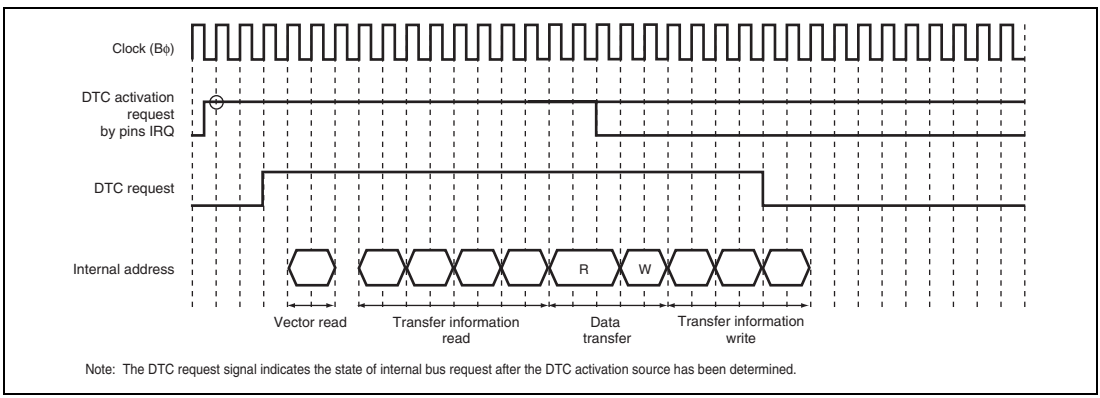
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.13 Example of DTC Operation Timing: Short Address Mode and Normal Transfer Mode or Repeat Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)



Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.14 Example of DTC Operation Timing: Normal Transfer, Repeat Transfer, DTPR=1 (Activated by On-Chip Peripheral Module; Iϕ: Bϕ: Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)



Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

Figure 8.15 Example of DTC Operation Timing: Normal Transfer, Repeat Transfer, (Activated by IRQ; Iϕ: Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)

8.5.8 Number of DTC Execution Cycles

Table 8.9 shows the execution status for a single DTC data transfer, and table 8.10 shows the number of cycles required for each execution.

Table 8.9 DTC Execution Status

Mode	Vector Read I		Transfer Information Read J		Transfer Information Write K			Data Read L	Data Write M	Internal Operation N	
Normal	1	0*1	4	0*1	3	2*2	1*3	1	1	1	0*1
Repeat	1	0*1	4	0*1	3	2*2	1*3	1	1	1	0*1
Block transfer	1	0*1	4	0*1	3	2*2	1*3	1•P	1•P	1	0*1

[Legend]

P: Block size (CRAH and CRAL value)

Notes: 1. When transfer information read is skipped
2. When the SAR or DAR is in fixed mode
3. When the SAR and DAR are in fixed mode

Table 8.10 Number of Cycles Required for Each Execution State

Object to be Accessed		On-Chip RAM*1	Flash Memory (ROM)	On-Chip I/O Registers*4			External Device*5		
Bus width		32 bits	32 bits	8 bits*4	16 bits	32 bits	8 bits	16 bits	32 bits
Access cycles		1Bφ to 4Bφ*1*2	3Bφ to 4Iφ + 3Bφ*2	2Pφ	2Pφ	2Pφ	2Bφ	2Bφ	2Bφ
Execution status	Vector read S_I	1Bφ to 4Bφ*1*2	3Bφ to 4Iφ + 3Bφ*2	—	—	—	9Bφ	5Bφ	3Bφ
	Transfer information read S_J	1Bφ to 4Bφ*1	—	—	—	—	9Bφ	5Bφ	3Bφ
	Transfer information write S_K	1Bφ to 3Bφ*1	—	—	—	—	2Bφ*6	2Bφ*6	2Bφ*6
	Byte data read S_L	1Bφ to 4Bφ*1	—	1Bφ + 2Pφ*3	1Bφ + 2Pφ*3	—	3Bφ	3Bφ	3Bφ
	Word data read S_L	1Bφ to 4Bφ*1	—	1Bφ + 2Pφ*3	1Bφ + 2Pφ*3	—	5Bφ	3Bφ	3Bφ
	Longword data read S_L	1Bφ to 4Bφ*1	—	1Bφ + 4Pφ*3	1Bφ + 2Pφ*3	1Bφ + 4Pφ*3	9Bφ	5Bφ	3Bφ
	Byte data write S_M	1Bφ to 3Bφ*1	—	1Bφ + 2Pφ*3	1Bφ + 2Pφ*3	—	2Bφ*6	2Bφ*6	2Bφ*6
	Word data write S_M	1Bφ to 3Bφ*1	—	1Bφ + 2Pφ*3	1Bφ + 2Pφ*3	—	2Bφ*6	2Bφ*6	2Bφ*6
	Longword data write S_M	1Bφ to 3Bφ*1	—	1Bφ + 4Pφ*3	1Bφ + 2Pφ*3	1Bφ + 4Pφ*3	2Bφ*6	2Bφ*6	2Bφ*6
	Internal operation S_N	1							

Notes: 1. Values for on-chip RAM. Number of cycles varies depending on the ratio of Iφ:Bφ.

	Read	Write
Iφ:Bφ = 1:1	3Bφ, 4Bφ	2Bφ, 3Bφ
Iφ:Bφ = 1:1/2	2Bφ, 3Bφ	2Bφ
Iφ:Bφ = 1:1/4	2Bφ	1Bφ, 2Bφ
Iφ:Bφ = 1:1/8	1Bφ	1Bφ

2. Values for the flash memory (ROM). Number of cycles varies depending on the ratio of Iϕ:Bϕ.

	Read
Iϕ:Bϕ = 1:1	4Iϕ + 3Bϕ
Iϕ:Bϕ = 1:1/2	4Iϕ + 3Bϕ
Iϕ:Bϕ = 1:1/4	4Iϕ + 3Bϕ
Iϕ:Bϕ = 1:1/8	3Bϕ

3. The values in the table are those for the fastest case. Depending on the state of the internal bus, replace 1Bϕ by 1Pϕ in a slow case.
4. The EtherC and E-DMAC are not included.
5. Values are different depending on the BSC register setting. The values in the table are the sample for the case with no wait cycles and the WM bit in CSnWCR = 1.
6. Values are different depending on the bus state.
   The number of cycles increases when many external wait cycles are inserted in the case where writing is frequently executed, such as block transfer, and when the external bus is in use because the write buffer cannot be used efficiently in such cases. For details on the write buffer, see section 9.5.12 (2), Access from the Side of the LSI Internal Bus Master.

The number of execution cycles is calculated from the formula below. Note that Σ means the sum of cycles for all transfers initiated by one activation event (the number of 1-valued CHNE bits in transfer information plus 1).

$$\text{Number of execution cycles} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S_N$$

8.5.9 DTC Bus Release Timing

The DTC requests the bus mastership of the internal bus (I bus) to the bus arbiter when an activation request occurs. The DTC releases the bus after a vector read, transfer information read, a single data transfer, or transfer information write-back. The DTC does not release the bus mastership during transfer information read, a single data transfer, or write-back of transfer information.

The bus release timing can be specified through the bus function extending register (BSCEHR). For details see section 9.4.8, Bus Function Extending Register (BSCEHR). The difference in bus release timing according to the register setting is summarized in table 8.11. Settings other than shown in the table are prohibited. The value of BSCEHR must not be modified while the DTC is active.

Figure 8.16 is a timing chart showing an example of bus release timing.

Table 8.11 DTC Bus Release Timing

	Bus Function Extending Register (BSCEHR) Setting		Bus Release Timing (O: Bus must be released; x: Bus is not released)				
						After Write-Back of Transfer Information	
	DTLOCK	DTBST	After Vector Read	After Transfer Information Read	After a Single data Transfer	Normal Transfer	Continuous Transfer
Setting 1	0	0	×	×	×	O	O
Setting 2*1	0	1	×	×	×	O	×
Setting 3*2	1	0	O	O	O	O	O

Notes: 1. The following restrictions apply to setting 2.
- The clock setting through the frequency control register (FRQCR) must be Iϕ : Bϕ : Pϕ : Mϕ : Aϕ = 16 : 4 : 4 : 4 : 4, 16 : 4 : 4 : 8 : 4, 8 : 4 : 4 : 8 : 4, or 8 : 4 : 4 : 4 : 4.
- The vector information must be stored in the flash memory (ROM) or on-chip RAM.
- The transfer information must be stored in the on-chip RAM.
- Transfer must be between the on-chip RAM and an on-chip peripheral module or between the external memory and an on-chip peripheral module.

2. The following restriction applies to setting 3.
- Use the DTPR bit in BSCEHR with this bit set to 0. Setting this bit to 1 is prohibited.

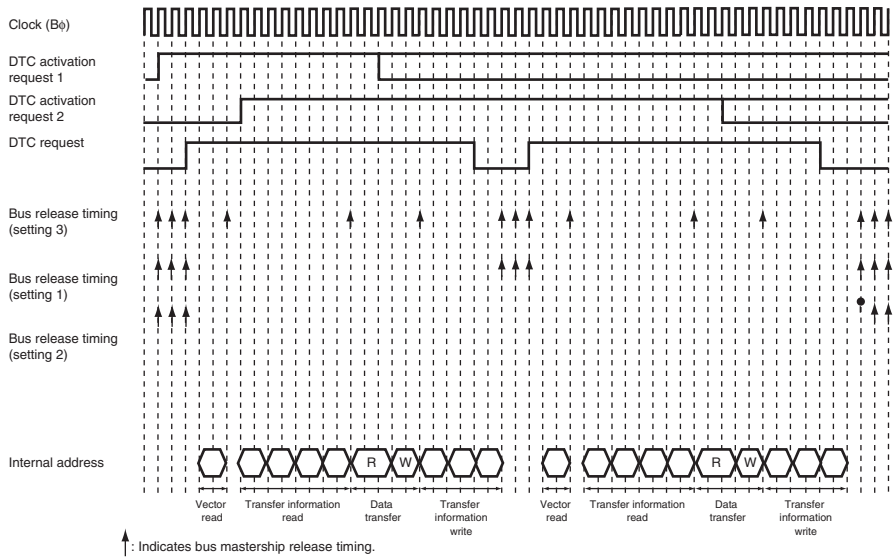


Figure 8.16 Example of DTC Operation Timing: Conflict of Two Activation Requests in Normal Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)

8.5.10 DTC Activation Priority Order

If multiple DTC activation requests are generated while the DTC is inactive, whether to start the DTC transfer from the first activation request* or according to the DTC activation priority can be selected through the DTPR bit setting in the bus function extending register (BSCEHR). If multiple activation requests are generated while the DTC is active, transfer is performed according to the DTC activation priority. Figure 8.17 shows an example of DTC activation according to the priority.

Note: * When one DTC-activation request is generated before another, transfer starts with the first request. When an activation request with a higher priority is generated before a pending DTC request is accepted, transfer starts for the request with higher priority. Timing of DTC request generation varies according to the operating state of internal buses.

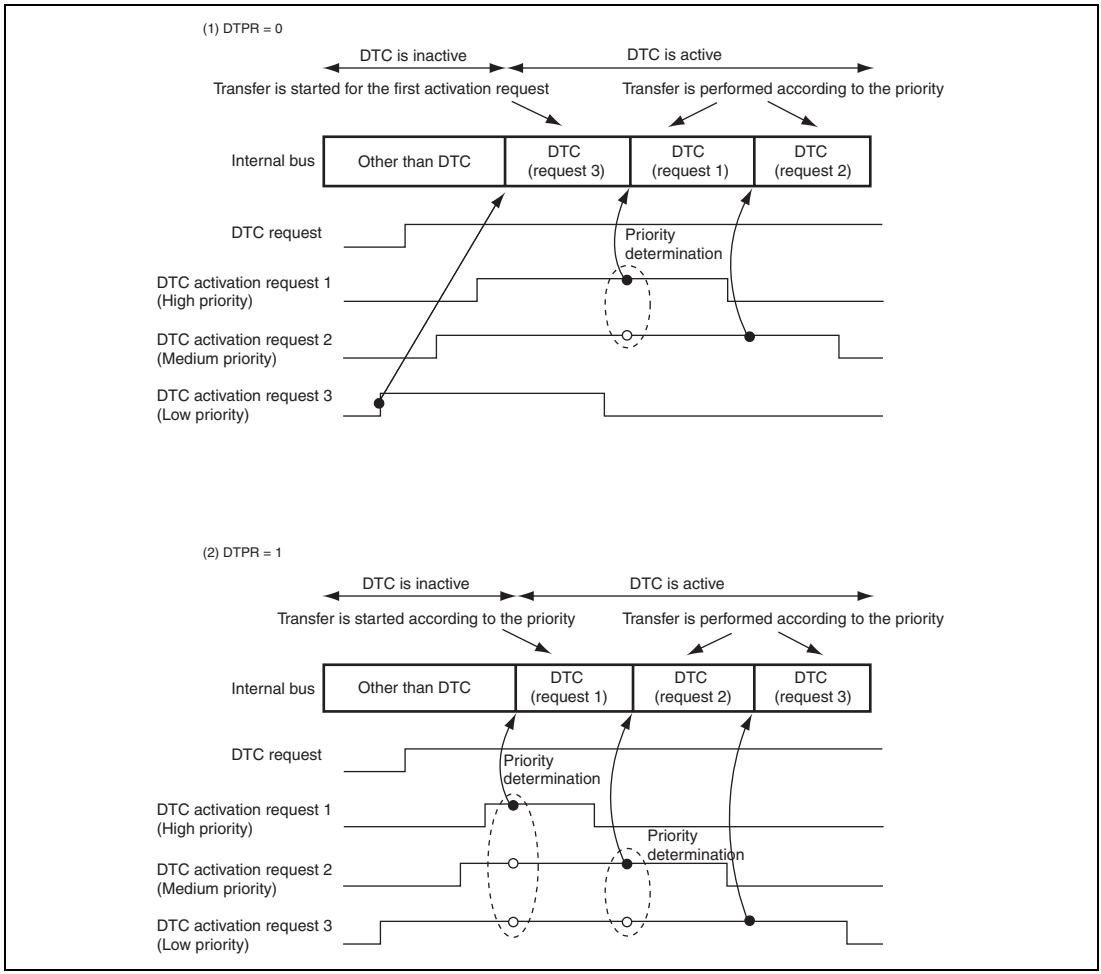
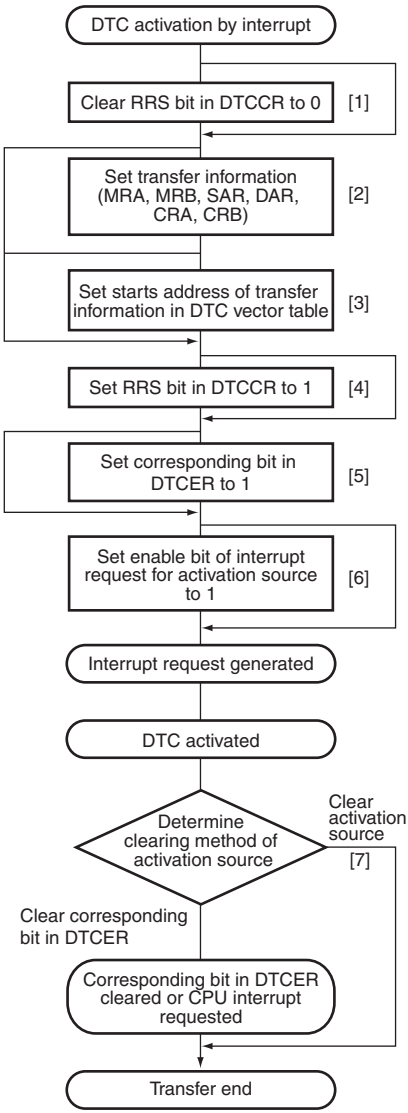


Figure 8.17 Example of DTC Activation According to Priority

8.6 DTC Activation by Interrupt

The procedure for using the DTC with interrupt activation is shown in figure 8.18.



[1] Clearing the RRS bit in DTCCR to 0 clears the read skip flag of transfer information. Read skip is not performed when the DTC is activated after clearing the RRS bit. When updating transfer information, the RRS bit must be cleared.

[2] Set the MRA, MRB, SAR, DAR, CRA, and CRB transfer information in the data area. For details on setting transfer information, see section 8.2, Register Descriptions. For details on location of transfer information, see section 8.4, Location of Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the DTC vector table. For details on setting DTC vector table, see section 8.4, Location of Transfer Information and DTC Vector Table.

[4] Setting the RRS bit to 1 performs a read skip of second time or later transfer information when the DTC is activated consecutively by the same interrupt source. Setting the RRS bit to 1 is always allowed. However, the value set during transfer will be valid from the next transfer.

[5] Set the bit in DTCER corresponding to the DTC activation interrupt source to 1. For the correspondence of interrupts and DTCER, refer to table 8.2. The bit in DTCER may be set to 1 on the second or later transfer. In this case, setting the bit is not needed.

[6] Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated. For details on the settings of the interrupt enable bits, see the corresponding descriptions of the corresponding module.

[7] After the end of one data transfer, the DTC clears the activation source flag or clears the corresponding bit in DTCER and requests an interrupt to the CPU. The operation after transfer depends on the transfer information. For details, see section 8.2, Register Descriptions and figure 8.4.

Figure 8.18 DTC Activation by Interrupt

8.7 Examples of Use of the DTC

8.7.1 Normal Transfer Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal transfer mode (MD1 = MD0 = 0), and byte size (Sz1 = Sz0 = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the RDR address of the SCI in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the transfer information for an RXI interrupt at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the receive end (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

8.7.2 Chain Transfer when Transfer Counter = 0

By executing a second data transfer and performing re-setting of the first data transfer only when the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-Kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 8.19 shows the chain transfer when the counter value is 0.

1. For the first transfer, set the normal transfer mode for input data. Set the fixed transfer source address, CRA = H'0000 (65,536 times), CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for 65,536-transfer units for the first data transfer in a separate area (in the flash memory (ROM), etc.). For example, if the input buffer is configured at addresses H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat transfer mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper eight bits of DAR in the first transfer information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.
4. Execute the first data transfer 65536 times by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
5. Next, execute the first data transfer the 65536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper eight bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
6. Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, no interrupt request is sent to the CPU.

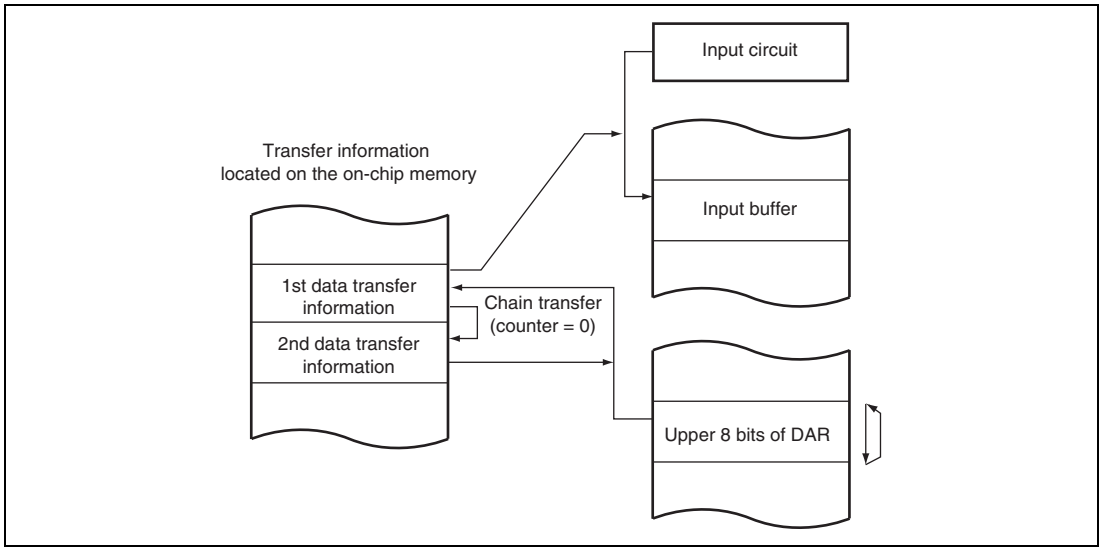


Figure 8.19 Chain Transfer when Transfer Counter = 0

8.8 Interrupt Sources

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or on completion of a single data transfer or a single block data transfer with the DISEL bit set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and priority level control in the interrupt controller. For details, refer to section 6.9, Data Transfer with Interrupt Request Signals.

8.9 Usage Notes

8.9.1 Module Standby Mode Setting

Operation of the DTC can be disabled or enabled using the standby control register. The initial setting is for operation of the DTC to be enabled. DTC operation and access are disabled in module standby mode. Do not place the DTC in module standby mode while it is active. Before entering software standby mode or module standby mode, all DTCER registers must be cleared. For details, refer to section 30, Power-Down Modes.

8.9.2 On-Chip RAM

Transfer information can be located in on-chip RAM. In this case, the corresponding RAME bits in SYSCR1 and SYSCR2 must not be cleared to 0.

8.9.3 DTCE Bit Setting

To set a DTCE bit, disable the corresponding interrupt, read 0 from the bit, and then write 1 to it. While DTC transfer is in progress, do not modify the DTCE bits.

8.9.4 Chain Transfer

When chain transfer is used, clearing of the activation source or DTCER is performed when the last of the chain of data transfers is executed. SCI, RSPI, RCAN-ET, SCIF, IIC3, USB and A/D converter interrupt/activation sources, on the other hand, are cleared when the DTC reads or writes to the relevant register during data transfer of the last of the chain.

Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.

8.9.5 Transfer Information Start Address, Source Address, and Destination Address

The transfer information start address to be specified in the vector table should be address 4n. Transfer information should be placed in on-chip RAM or external memory space.

8.9.6 Access to DTC Registers through DTC

Do not access the DMAC or DTC registers by using DTC operation. Do not access the DTC registers by using DMAC operation.

8.9.7 Notes on IRQ Interrupt as DTC Activation Source

When a low level on the IRQ pin is to be detected, if the end of DTC transfer is used to request an interrupt to the CPU (transfer counter = 0 or DISEL = 1), the IRQ signal must be held low until the CPU accepts the interrupt.

8.9.8 Note on SCI or SCIF as DTC Activation Sources

When the TXI interrupt from the SCI is specified as a DTC activation source, the TEND flag in the SCI must not be used as the transfer end flag.

When the TXIF interrupt from the SCIF is specified as a DTC activation source, the TEND flag in the SCIF must not be used as the transfer end flag.

8.9.9 Clearing Interrupt Source Flag

The interrupt source flag set when the DTC transfer is completed should be cleared in the interrupt handler in the same way as for general interrupt source flags. For details, refer to section 6.10, Usage Note.

8.9.10 Conflict between NMI Interrupt and DTC Activation

When a conflict occurs between the generation of the NMI interrupt and the DTC activation, the NMI interrupt has priority. Thus the ERR bit is set to 1 and the DTC is not activated.

It takes 3Bϕ + 2Pϕ for checking DTC stop by the NMI, 3Bϕ + 2Pϕ for checking DTC activation by the IRQ, and 1Bϕ + 1Pϕ to 4Bϕ + 1Pϕ for checking DTC activation by the peripheral module.

8.9.11 Note on USB as DTC Activation Sources

To generate a CPU interrupt when a DTC transfer activated by the USB is completed, refer to the procedure described in section 24, USB Function Module (USB).

8.9.12 Operation when a DTC Activation Request has been Cancelled

Once DTC has accepted an activation request, the next activation request will not be accepted until the sequence of the DTC transaction has finished up to the end of write-back.

8.9.13 Note on Writing to DTCER

When the same condition has been set as both a DTC activation source and a CPU interrupt source, if the interrupt as both sources is generated while DTCER is also set for the DTC activation source, the DTC and CPU may be activated at the same time. Determine the value of DTCER before allowing the generation of DTC activation interrupts.

Section 9 Bus State Controller (BSC)

The bus state controller (BSC) outputs control signals for various types of memory that is connected to the external address space and external devices. BSC functions enable this LSI to connect directly with SRAM, SDRAM, and other memory storage devices, and external devices.

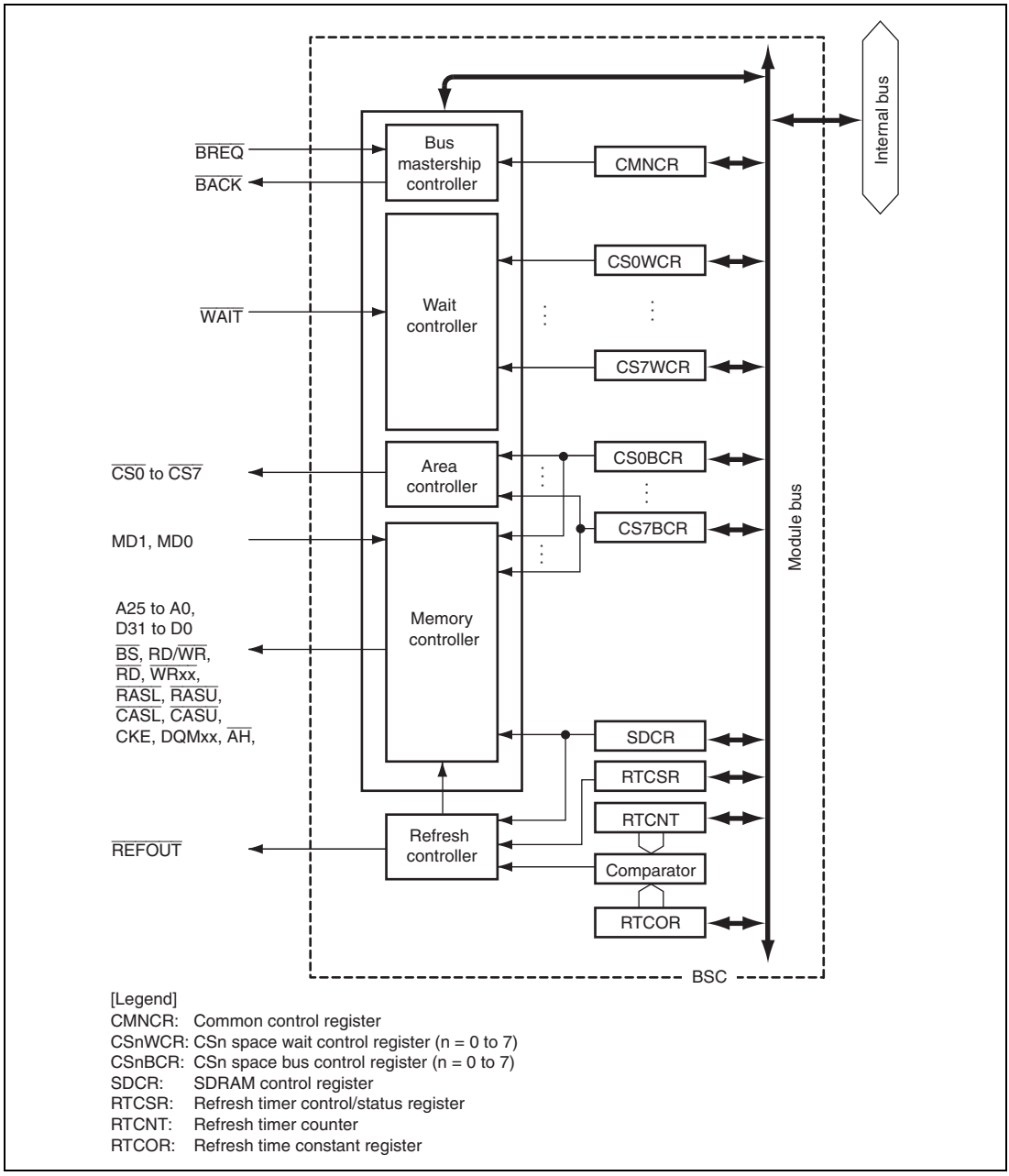
9.1 Features

The BSC has the following features.

1. External address space
   - A maximum of 64 Mbytes for each of areas CS0 to CS7.
   - Can specify the normal space interface, SRAM interface with byte selection, burst ROM (clock synchronous or asynchronous), MPX-I/O, and SDRAM for each address space.
   - Can select the data bus width (8, 16, or 32 bits) for each address space.
   - Controls insertion of wait cycles for each address space.
   - Controls insertion of wait cycles for each read access and write access.
   - Can set independent idle cycles during the continuous access for five cases: read-write (in same space/different spaces), read-read (in same space/different spaces), the first cycle is a write access.
2. Normal space interface
   - Supports the interface that can directly connect to the SRAM.
3. Burst ROM interface (clock asynchronous)
   - High-speed access to the ROM that has the page mode function.
4. MPX-I/O interface
   - Can directly connect to a peripheral LSI that needs an address/data multiplexing.
5. SDRAM interface
   - Can set the SDRAM in up to two areas.
   - Multiplex output for row address/column address.
   - Efficient access by single read/single write.
   - High-speed access in bank-active mode.
   - Supports an auto-refresh and self-refresh.
   - Supports low-frequency and power-down modes.
   - Issues MRS and EMRS commands.

6. SRAM interface with byte selection
   — Can connect directly to a SRAM with byte selection.
7. Burst ROM interface (clock synchronous)
   — Can connect directly to a ROM of the clock-synchronous type.
8. Bus arbitration
   — Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.
9. Refresh function
   — Supports the auto-refresh and self-refresh functions.
   — Specifies the refresh interval using the refresh counter and clock selection.
   — Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8).
10. Usage as interval timer for refresh counter
   — Generates an interrupt request at compare match.

Figure 9.1 shows a block diagram of the BSC.



[Legend]
CMNCR: Common control register
CSnWCR: CSn space wait control register (n = 0 to 7)
CSnBCR: CSn space bus control register (n = 0 to 7)
SDCR: SDRAM control register
RTCSR: Refresh timer control/status register
RTCNT: Refresh timer counter
RTCOR: Refresh time constant register

Figure 9.1 Block Diagram of BSC

9.2 Input/Output Pins

Table 9.1 shows the pin configuration of the BSC.

Table 9.1 Pin Configuration

Name	I/O	Function
A25 to A0	Output	Address bus
D31 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$ to $\overline{CS7}$	Output	Chip select
RD/$\overline{WR}$	Output	Read/write Connects to $\overline{WE}$ pins when SDRAM or SRAM with byte selection is connected.
$\overline{RD}$	Output	Read pulse signal (read data output enable signal) Functions as a strobe signal for indicating memory read cycles when PCMCIA is used.
$\overline{AH}$	Output	A signal used to hold an address when MPX-I/O is in use
WRHH/DQMUU	Output	Indicates that D31 to D24 are being written to. Connected to the byte select signal when SRAM with byte selection is connected. Functions as the select signals for D31 to D24 when SDRAM is connected.
WRHL/DQMUL	Output	Indicates that D23 to D26 are being written to. Connected to the byte select signal when SRAM with byte selection is connected. Functions as the select signals for D23 to D26 when SDRAM is connected.
WRH/DQMLU	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D15 to D8 when SDRAM is connected.

Name	I/O	Function
WRL/DQMLL	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D7 to D0 when SDRAM is connected.
$\overline{\text{RASL}}$, $\overline{\text{RASU}}$	Output	Connected to $\overline{\text{RAS}}$ pin when SDRAM is connected.
$\overline{\text{CASL}}$, $\overline{\text{CASU}}$	Output	Connected to $\overline{\text{CAS}}$ pin when SDRAM is connected.
CKE	Output	Connected to CKE pin when SDRAM is connected.
$\overline{\text{WAIT}}$	Input	External wait input
$\overline{\text{BREQ}}$	Input	Bus request input
$\overline{\text{BACK}}$	Output	Bus enable output
$\overline{\text{REFOUT}}$	Output	Refresh request output in bus-released state
MD0	Input	Selects bus width (16 or 32 bits) of area 0. It also selects the on-chip ROM enabled or disabled mode and external bus access enabled or disabled mode.

9.3 Area Overview

9.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into external address space and on-chip spaces (on-chip ROM, on-chip RAM, on-chip peripheral modules, and reserved areas) according to the upper bits of the address.

The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is listed below.

Table 9.2 Address Map in On-Chip ROM-Enabled Mode

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'000F FFFF	On-chip ROM	On-chip ROM	512 kbytes (SH72165, SH72145), 768 kbytes (SH72166, SH72146), 1 Mbyte (SH72167, SH72147)
H'0070 0000 to H'01FF FFFF	Other	Reserved area	—
H'0200 0000 to H'03FF FFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)	32 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM, reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 29, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 32, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

Table 9.3 Address Map in On-Chip ROM-Disabled Mode

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'03FF FFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)	64 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM, reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 29, On-Chip RAM. For the on-chip I/O register space, access the addresses shown in section 32, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

9.3.2 Setting Operating Modes

This LSI can set the following modes of operation at the time of power-on reset using the external pins.

- Single-Chip Mode/External Bus Accessible Mode

  In single-chip mode, no access is made to the external bus, and the LSI is activated by the on-chip ROM program upon a power-on reset. The BSC module enters the module standby state to reduce power consumption.

  The address, data, bus control pins used in external bus accessible mode can be used as the port function pins in single-chip mode.

- On-Chip ROM-Enabled Mode/On-Chip ROM-Disabled Mode

  In on-chip ROM-enabled mode, since the first half of area 0 is allocated to the on-chip ROM, the LSI can be activated by the on-chip ROM program upon a power-on reset. The second half of area 0 is the external memory space.

  In on-chip ROM-disabled mode, the LSI is activated by the program stored in the external memory allocated to area 0. The second half of area 0 is the external memory space. In this case, a ROM is assumed for the external memory of area 0. Therefore, minimum functions are provided for the pins including address bus, data bus, CS0, and RD. Although $\overline{\text{BS}}$, RD/$\overline{\text{WR}}$, $\overline{\text{WR}}$xx, and other pins are shown in the examples of access waveforms in this section, these are examples when pin settings are performed by the pin function controller. For details, see section 22, Pin Function Controller (PFC). Do not perform any operation except for area 0 read access until the pin settings by the program is completed.

- Initial Settings of Data Bus Widths for Areas 0 to 7

  The initial settings of data bus widths of areas 0 to 7 can be selected at a time as 16 bits or 32 bits.

  In on-chip ROM-disabled mode, the data bus width of area 0 cannot be changed from its initial setting after a power-on reset, but the data bus widths of areas 1 to 7 can be changed by register settings in the program. In on-chip ROM-enabled mode, all the data bus widths of areas 0 to 7 can be changed by register settings in the program. Note that data bus widths will be restricted depending on memory types.

- Initial Settings of Big Endian / Little Endian

  The initial settings of byte-data alignment of areas 1 to 7 can be selected as big endian or little endian. In on-chip ROM-disabled mode, the endianness of area 0 cannot be changed from its initial setting after a power-on reset, but the endianness of areas 1 to 7 can be changed by register settings in the program. In on-chip ROM-enabled mode, all the endianness of areas 1 to 7 can be changed by register settings in the program. Area 0 cannot be selected as little endian. Since the instruction fetch is mixed with the 32- and 16-bit access and the allocation to the little endian area is difficult, the instruction must be executed within the big endian area.

For details of mode settings, see section 3, MCU Operating Modes.

9.4 Register Descriptions

The BSC has the following registers.

Do not access spaces other than area 0 until settings of the connected memory interface are completed.

Table 9.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	CMNCR	R/W	H'00001010	H'FFFC0000	32
CSn space bus control register	CSnBCR	R/W	H'36DB0400*	H'FFFC 0004 to H'FFFC 0020	32
CSn space wait control register	CSnWCR	R/W	H'00000500	H'FFFC0028 to H'FFFC 0044	32
SDRAM control register	SDCR	R/W	H'00000000	H'FFFC004C	32
Refresh timer control/status register	RTCSR	R/W	H'00000000	H'FFFC0050	32
Refresh timer counter	RTCNT	R/W	H'00000000	H'FFFC0054	32
Refresh time constant register	RTCOR	R/W	H'00000000	H'FFFC0058	32
Bus function extending register	BSCEHR	R/W	H'0000	H'FFFE3C1A	16

Note: * Value when selecting the16-bit bus width with the external pin (MD0). When selecting the 32-bit bus width, the initial value will be H'36DB 0600.

9.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. This register is initialized to H'00001010 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	BLOCK	DPRTY[1:0]		DMAIW[2:0]			DMA IWA	-	-	HIZ CKIO	HIZ MEM	HIZ CNT
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
11	BLOCK	0	R/W	Bus Lock Specifies whether or not the $\overline{\text{BREQ}}$ signal is received. 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.
10, 9	DPRTY[1:0]	00	R/W	DMA Burst Transfer Priority Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer. 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer. 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer. 11: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	DMAIW[2:0]	000	R/W	Wait states between access cycles when DMA single address transfer is performed. Specify the number of idle cycles to be inserted after an access to an external device with DACK when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
5	DMAIWA	0	R/W	Method of inserting wait states between access cycles when DMA single address transfer is performed. Specifies the method of inserting the idle cycles specified by the DMAIW[2:0] bit. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with DACK drove it. However, when the external device with DACK drives the data bus continuously, idle cycles are not inserted. Setting this bit will make this LSI insert the idle cycles after an access to an external device with DACK, even when the continuous access cycles to an external device with DACK are performed. 0: Idle cycles inserted when another device drives the data bus after an external device with DACK drove it. 1: Idle cycles always inserted after an access to an external device with DACK
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	HIZCKIO	0	R/W	High-Z CK Control Specifies the state in CK standby mode and when bus mastership is released. 0: CK is in high impedance state in standby mode and bus-released state. 1: CK is driven in standby mode and bus-released state.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in standby mode for A25 to A0, $\overline{BS}$, $\overline{CSn}$, RD/$\overline{WR}$, $\overline{WRxx}$/DQMxx, $\overline{AH}$, and $\overline{RD}$. At bus-released state, these pins are in high-impedance state regardless of the setting value of the HIZMEM bit. 0: High impedance in standby mode. 1: Driven in standby mode
0	HIZCNT	0	R/W	High-Z Control Specifies the state in standby mode and bus-released state for CKE, $\overline{RASL}$, $\overline{CASL}$, $\overline{RASU}$, and $\overline{CASU}$. 0: CKE, $\overline{RASL}$, $\overline{CASL}$, $\overline{RASU}$, and $\overline{CASU}$ are in high-impedance state in standby mode and bus-released state. 1: CKE, $\overline{RASL}$, $\overline{CASL}$, $\overline{RASU}$, and $\overline{CASU}$ are driven in standby mode and bus-released state.

9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7)

CSnBCR is a 32-bit readable/writable register that specifies the type of memory connected to a space, data bus width of an area, endian, and the number of waits between access cycles. This register is initialized to H'36DB0x00 by a power-on reset and retains the value by a manual reset and in software standby mode.

Do not access external memory other than area 0 until CSnBCR initial setting is completed.

Idle cycles may be inserted even when they are not specified. For details, see section 9.5.10, Wait between Access Cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Initial value:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TYPE[2:0]			ENDIAN	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0*	1*	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	IWW[2:0]	011	R/W	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycles are the write-read cycle and write-write cycle. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
27 to 25	IWRWD[2:0]	011	R/W	Idle Cycles for Another Space Read-Write Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target access cycle is a read-write one in which continuous access cycles switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
24 to 22	IWRWS[2:0]	011	R/W	Idle Cycles for Read-Write in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-write cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted

Bit	Bit Name	Initial Value	R/W	Description
21 to 19	IWRRD[2:0]	011	R/W	Idle Cycles for Read-Read in Another Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles switch between different spaces. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
18 to 16	IWRRS[2:0]	011	R/W	Idle Cycles for Read-Read in the Same Space Specify the number of idle cycles to be inserted after the access to a memory that is connected to the space. The target cycle is a read-read cycle of which continuous access cycles are for the same space. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	TYPE[2:0]	000	R/W	Specify the type of memory connected to a space. 000: Normal space 001: Burst ROM (clock asynchronous) 010: MPX-I/O 011: SRAM with byte selection 100: SDRAM 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Burst ROM (clock synchronous) For details of memory type in each area, see tables 9.2 and 9.3.
11	ENDIAN	0	R/W	Endian Select Specifies data alignment in a space. 0: Big endian 1: Little endian

Bit	Bit Name	Initial Value	R/W	Description
10, 9	BSZ[1:0]	01*	R/W	Data Bus Width Specification Specify the data bus widths of spaces. 00: Reserved (setting prohibited) 01: 8-bit size 10: 16-bit size 11: 32-bit size For MPX-I/O, selects bus width by address. Notes: 1. If area 5 is specified as MPX-I/O, the bus width can be specified as 8 bits or 16 bits by the address according to the SZSEL bit in CS5WCR by specifying the BSZ[1:0] bits to 11. The fixed bus width can be specified as 8 bits or 16 bits. 2. The initial data bus width for areas 0 to 7 is specified by external pins. In on-chip ROM-disabled mode, writing to the BSZ1 and BSZ0 bits in CS0BCR is ignored, but the bus width settings in CS1BCR to CS7BCR can be modified. In on-chip ROM-enabled mode, the bus width settings in CS0BCR to CS7BCR can be modified. 3. If area 0 or 4 is specified as clock-synchronous burst ROM space, the bus width can be specified as 16 bits only.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * Details of Initial value of this bit are shown below according to the MCU operating mode.
Initial value in mode 0: B'11
Initial value in mode 1: B'10
Initial value in mode 2: B'01
Initial value in mode 3: B'00

9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7)

CSnWCR specifies various wait cycles for memory access. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

CSnWCR is initialized to H'00000500 by a power-on reset and retains the value by a manual reset and in software standby mode.

(1) Normal Space, SRAM with Byte Selection, MPX-I/O

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	— *	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
20	BAS*	0	R/W	Byte Access Selection when SRAM with Byte Selection is Used Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read/write timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read/write access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19 to 13	— *	All 0	R/W	Reserved Set these bits to 0 when the interface for normal space or SRAM with byte selection is used.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{\text{CS0}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CS0}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WRxx}}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS0}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Note * To connect the burst ROM to the CS0 space and switch to the burst ROM interface after activation in ROM-disabled mode, set the TYPE[2:0] bits in CS0BCR after setting the burst number by the bits 20 and 21 and the burst wait cycle number by the bits 16 and 17. Do not write 1 to the reserved bits other than above bits.

- CS1WCR, CS7WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WRxx}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WRxx}$ signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WRxx}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.

Bit	Bit Name	Initial Value	R/W	Description
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{\text{CSn}}$ Assertion to $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ Assertion Specify the number of delay cycles from address and $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$ and $\overline{\text{WRxx}}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{\text{WRxx}}$ Negation to Address, $\overline{\text{CSn}}$ Negation Specify the number of delay cycles from RD and $\overline{\text{WRxx}}$ negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS2WCR, CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	WR[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS5WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SZSEL	MPXW/ BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
21	SZSEL	0	R/W	MPX-I/O Interface Bus Width Specification Specifies an address to select the bus width when the BSZ[1:0] of CS5BCR are specified as 11. This bit is valid only when area 5 is specified as MPX-I/O. 0: Selects the bus width by address A14 1: Selects the bus width by address A21 The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized below.

SZSEL	A14	A21	Bus Width
0	0	Not affected	8 bits
0	1	Not affected	16 bits
1	Not affected	0	8 bits
1	Not affected	1	16 bits

Bit	Bit Name	Initial Value	R/W	Description
20	MPXW	0	R/W	MPX-I/O Interface Address Wait This bit setting is valid only when area 5 is specified as MPX-I/O. Specifies the address cycle insertion wait for MPX-I/O interface. 0: Inserts no wait cycle 1: Inserts 1 wait cycle
	BAS	0	R/W	SRAM with Byte Selection Byte Access Select This bit setting is valid only when area 5 is specified as SRAM with byte selection. Specifies the $\overline{\text{WRxx}}$ and RD/$\overline{\text{WR}}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{\text{WRxx}}$ signal at the read timing and asserts the RD/$\overline{\text{WR}}$ signal during the write access cycle. 1: Asserts the $\overline{\text{WRxx}}$ signal during the read access cycle and asserts the RD/$\overline{\text{WR}}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS5}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion when area 5 is specified as normal space or SRAM with byte selection. Specify the number of delay cycles from the end of address cycle (Ta3) to $\overline{RD}$ and $\overline{WRxx}$ assertion when area 5 is specified as MPx-I/O. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS5}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS5}$ negation when area 5 is specified as normal space or SRAM with byte selection. Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to $\overline{CS5}$ negation when area 5 is specified as MPx-I/O. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS6WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WRxx}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WRxx}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WRxx}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.

Bit	Bit Name	Initial Value	R/W	Description
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS6}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address, $\overline{CS6}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WN	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification of this bit is valid even when the number of access wait cycles is 0. 0: The external wait input is valid 1: The external wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRxx}$ negation to address, and $\overline{CS6}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(2) Burst ROM (Clock Asynchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11.

Bus Width	BST[1:0]	Burst count
8 bits	00	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst

Bit	Bit Name	Initial Value	R/W	Description
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		W[3:0]				WM	-	-	-	-	HW[1:0]	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bus Width	BST[1:0]	Burst count
8 bits	00	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WRxx}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WRxx}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRxx}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WRxx}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(3) SDRAM*

- CS2WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	A2CL[1:0]		-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A2CL[1:0]	10	R/W	CAS Latency for Area 2 Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

- CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	WTRP[1:0]*		-	WTRCD[1:0]*		-	A3CL[1:0]		-	-	TRWL[1:0]*		-	WTRC[1:0]*	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14, 13	WTRP[1:0]*	00	R/W	Number of Auto-Precharge Completion Wait Cycles Specify the number of minimum precharge completion wait cycles as shown below. • From the start of auto-precharge and issuing of ACTV command for the same bank • From issuing of the PRE/PALL command to issuing of the ACTV command for the same bank • Till entering power-down mode or deep power-down mode • From the issuing of PALL command to issuing REF command in auto-refresh mode • From the issuing of PALL command to issuing SELF command in self-refresh mode The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11, 10	WTRCD[1:0]*	01	R/W	Number of Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command Specify the minimum number of wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A3CL[1:0]	10	R/W	CAS Latency for Area 3 Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	TRWL[1:0]*	00	R/W	Number of Auto-Precharge Startup Wait Cycles Specify the number of minimum auto-precharge startup wait cycles as shown below. • Cycle number from the issuance of the WRITA command by this LSI until the completion of auto-precharge in the SDRAM. Equivalent to the cycle number from the issuance of the WRITA command until the issuance of the ACTV command. Confirm that how many cycles are required between the WRITE command receive in the SDRAM and the auto-precharge activation, referring to each SDRAM data sheet. And set the cycle number so as not to exceed the cycle number specified by this bit. • Cycle number from the issuance of the WRITA command until the issuance of the PRE command. This is the case when accessing another low address in the same bank in bank active mode. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	WTRC[1:0]*	00	R/W	Number of Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the periods shown below. • From the issuance of the REF command until the issuance of the ACTV/REF/MRS command • From releasing self-refresh until the issuance of the ACTV/REF/MRS command. The setting for areas 2 and 3 is common. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.
If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

(4) Burst ROM (Clock Synchronous)

- CS0WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BW[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

SDCR is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	A2ROW[1:0]		-	A2COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV	-	-	-	A3ROW[1:0]		-	A3COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20, 19	A2ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 2 Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RFSH or RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address, and write data for SDRAM and the latch timing of read data from SDRAM. Setting this bit makes the hold time for command, address, write and read data extended for half cycle (output or read at the falling edge of CK). This mode is suitable for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM is output at the rising edge of CK. Read data from SDRAM is latched at the rising edge of CK. 1: Command, address, and write data for SDRAM is output at the falling edge of CK. Read data from SDRAM is latched at the falling edge of CK.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh

Bit	Bit Name	Initial Value	R/W	Description
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed
9	PDOWN	0	R/W	Power-Down Mode Specifies whether the SDRAM will enter power-down mode after the access to the SDRAM. With this bit being set to 1, after the SDRAM is accessed, the CKE signal is driven low and the SDRAM enters power-down mode. 0: The SDRAM does not enter power-down mode after being accessed. 1: The SDRAM enters power-down mode after being accessed.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be set only in area 3, and only the 16-bit bus width can be set. When both the CS2 and CS3 spaces are set to SDRAM, specify auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	A3ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 3 Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	A3COL[1:0]	00	R/W	Number of Bits of Column Address for Area 3 Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

9.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM. RTCSR is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

When RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

The phase of the clock for incrementing the count in the refresh timer counter (RTCNT) is adjusted only by a power-on reset. Note that there is an error in the time until the compare match flag is set for the first time after the timer is started with the CKS[2:0] bits being set to a value other than B'000.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	CKS[2:0]			RRC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.

Bit	Bit Name	Initial Value	R/W	Description
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables CMF interrupt requests when the CMF bit in RTCSR is set to 1. 0: Disables CMF interrupt requests. 1: Enables CMF interrupt requests.
5 to 3	CKS[2:0]	000	R/W	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2 to 0	RRC[2:0]	000	R/W	Refresh Count Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long. 000: 1 time 001: 2 times 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

9.4.6 Refresh Timer Counter (RTCNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS[2:0] in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This counter is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Counter

9.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

The $\overline{\text{REFOUT}}$ signal can be asserted when a refresh request is generated while the bus is released. For details, see the description of Relationship between Refresh Requests and Bus Cycles in section 9.5.6 (9), Relationship between Refresh Requests and Bus Cycles, and section 9.5.11, Bus Arbitration.

When the CMIE bit in RTCSR is set to 1, an interrupt request is issued by this matching signal. The request continues to be output until the CMF bit in RTCSR is cleared. Clearing the CMF bit only affects the interrupt request and does not clear the refresh request. Therefore, a combination of refresh request and interval timer interrupt can be specified so that the number of refresh requests are counted by using timer interrupts while refresh is performed periodically.

When RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This register is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Register

9.4.8 Bus Function Extending Register (BSCEHR)

BSCEHR is a 16-bit register that specifies the timing of DTC or DMAC bus release. It is used to give priority to DTC or DMAC transfer or reduce the number of cycles in which the DTC is active.

For the differences in DTC operation according to the combinations of the DTLOCK and DTBST bit settings, refer to section 8.5.9, DTC Bus Release Timing.

Setting the DTSA bit enables DTC short address mode. For details of the short address mode, see section 8.4, Location of Transfer Information and DTC Vector Table.

The DTPR bit selects the DTC activation priority used when multiple DTC activation requests are generated before DTC activation.

Do not modify this register while the DMAC or DTC is active.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT LOCK	-	-	-	DTBST	DTSA	-	DTPR	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	DTLOCK	0	R/W	DTC Lock Enable Specifies the timing of DTC bus release. 0: The DTC releases the bus when the NOP instruction is issued after vector read, or after write-back of transfer information is completed. 1: The DTC releases the bus after vector read, when the NOP instruction is issued after vector read, after transfer information read, after a single data transfer, or after write-back of transfer information.
14 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11	DTBST	0	R/W	DTC Burst Enable Selects whether the DTC continues operation without releasing the bus when multiple DTC activation requests are generated. 0: The DTC releases the bus every time a DTC activation request has been processed. 1: The DTC continues operation without releasing the bus until all DTC activation requests have been processed. Notes: When this bit is set to 1, the following restrictions apply. 1. Clock setting through the frequency control register (FRQCR) must be Iϕ : Bϕ : Pϕ: Mϕ: Aϕ = 16 : 4 : 4 : 4 : 4, 16 : 4 : 4 : 8 : 4, 8 : 4 : 4 : 4 : 4, or 8 : 4 : 4 : 8 : 4 2. The vector information must be stored in the on-chip ROM or on-chip RAM. 3. The transfer information must be stored in the on-chip RAM. 4. Transfer must be between the on-chip RAM and an on-chip peripheral module or between the external memory and an on-chip peripheral module. 5. Do not set the DTBST bit to 1, when the activation source is low-level setting for IRQ7 to IRQ0 and the RRS bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
10	DTSA	0	R/W	DTC Short Address Mode Selects the short address mode in which only three longwords are required for DTC transfer information read. 0: Four longwords are read as the transfer information. The transfer information is arranged as shown in the figure for normal mode in figure 8.2. 1: Three longwords are read as the transfer information. The transfer information is arranged as shown in the figure for short address mode in figure 8.2. Note: The short address mode can be used only for transfer between an on-chip peripheral module and the on-chip RAM because the upper eight bits of SAR and DAR are assumed as all 1s.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	DTPR	0	R/W	DTC Activation Priority Selects whether to start transfer from the first DTC activation request or according to the DTC activation priority when multiple DTC activation requests are generated before the DTC is activated. For details, see section 8.5.10, DTC Activation Priority Order. 0: Starts transfer from the DTC activation request generated first. 1: Starts transfer according to the DTC activation priority. Notes: When this bit is set to 1, the following restrictions apply. 1. The vector information must be stored in the on-chip ROM or on-chip RAM. 2. The transfer information must be stored in the on-chip RAM. 3. The function for skipping the transfer information read step is always disabled. 4. Set this bit to 1 while DTLOCK = 0. The DTLOCK bit should not be set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.5 Operation

9.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian in which the 0 address is the most significant byte (MSB), and little endian in which the 0 address is the least significant byte (LSB) in the byte data. In a space of areas 1 to 7, endian can be set by the CSnBCR setting while the target space is not accessed. In a space of area 0, the CSnBCR setting is invalid in on-chip ROM-disabled mode. In on-chip ROM-enabled mode, endian can be set by the CSnBCR setting in a space of areas 0 to 7.

For normal memory and SRAM with byte selection, the data bus width can be selected from three widths (8, 16, and 32 bits). For SDRAM, the data bus width can be selected from two widths (16 and 32 bits). For MPX-I/O, the data bus width is fixed at 8 bits or 16 bits, or 8 bits or 16 bits can be selected by the access address. Data alignment is performed in accordance with the data bus width of the device. This also means that when longword data is read from a byte-width device, the read operation must be done four times. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Tables 9.5 to 9.10 show the relationship between device data width and access unit. Note that addresses corresponding to the strobe signals for the 16-bit bus width differ between big endian and little endian. $\overline{\text{WRH}}$ indicates the 0 address in big-endian mode, but $\overline{\text{WRL}}$ indicates the 0 address in little-endian mode.

Area 0 cannot be selected as little endian. Since the instruction fetch is mixed with the 32- and 16-bit access and the allocation to the little endian area is difficult, the instruction must be executed within the big endian area.

Table 9.5 32-Bit External Device Access and Data Alignment in Big-Endian Mode

	Data Bus				Strobe Signals			
Operation	D31 to D24	D23 to D16	D15 to D8	D7 to D0	$\overline{\text{WRHH}}$, DQMUU	$\overline{\text{WRHL}}$, DQMUL	$\overline{\text{WRH}}$, DQMLU	$\overline{\text{WRL}}$, DQMLL
Byte access at 0	Data 7 to 0	—	—	—	Assert	—	—	—
Byte access at 1	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 2	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 3	—	—	—	Data 7 to 0	—	—	—	Assert
Word access at 0	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Word access at 2	—		Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 9.6 16-Bit External Device Access and Data Alignment in Big-Endian Mode

		Data Bus		Strobe Signals	
Operation		D15 to D8	D7 to D0	$\overline{\text{WRH}}$, DQMLU	$\overline{\text{WRL}}$, DQMLL
Byte access at 0		Data 7 to 0	—	Assert	—
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		Data 7 to 0	—	Assert	—
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword access at 0	1st time at 0	Data 23 to 16	Data 31 to 24	Assert	Assert
	2nd time at 2	Data 7 to 0	Data 15 to 8	Assert	Assert

Table 9.7 8-Bit External Device Access and Data Alignment in Big-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{\text{WRH}}$, DQMLU**	**$\overline{\text{WRL}}$, DQMLL**
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0	1st time at 0	—	Data 15 to 8	—	Assert
	2nd time at 1	—	Data 7 to 0	—	Assert
Word access at 2	1st time at 2	—	Data 15 to 8	—	Assert
	2nd time at 3	—	Data 7 to 0	—	Assert
Longword access at 0	1st time at 0	—	Data 31 to 24	—	Assert
	2nd time at 2	—	Data 23 to 16	—	Assert
	3rd time at 2	—	Data 15 to 8	—	Assert
	4th time at 3	—	Data 7 to 0	—	Assert

Table 9.8 32-Bit External Device Access and Data Alignment in Little-Endian Mode

	Data Bus				Strobe Signals			
Operation	**D31 to D24**	**D23 to D16**	**D15 to D8**	**D7 to D0**	**$\overline{\text{WRHH}}$, DQMUU**	**$\overline{\text{WRHL}}$, DQMUL**	**$\overline{\text{WRH}}$, DQMLU**	**$\overline{\text{WRL}}$, DQMLL**
Byte access at 0	—	—	—	Data 7 to 0	—	—	—	Assert
Byte access at 1	—	—	Data 7 to 0	—	—	—	Assert	—
Byte access at 2	—	Data 7 to 0	—	—	—	Assert	—	—
Byte access at 3	Data 7 to 0	—	—	—	Assert	—	—	—
Word access at 0	—	—	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert
Word access at 2	Data 15 to 8	Data 7 to 0	—	—	Assert	Assert	—	—
Longword access at 0	Data 31 to 24	Data 23 to 16	Data 15 to 8	Data 7 to 0	Assert	Assert	Assert	Assert

Table 9.9 16-Bit External Device Access and Data Alignment in Little-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{\text{WRH}}$, DQMLU**	**$\overline{\text{WRL}}$, DQMLL**
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		Data 7 to 0	—	Assert	—
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		Data 7 to 0	—	Assert	—
Word access at 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword access at 0	1st time at 0	Data 15 to 8	Data 7 to 0	Assert	Assert
	2nd time at 2	Data 31 to 24	Data 23 to 16	Assert	Assert

Table 9.10 8-Bit External Device Access and Data Alignment in Little-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{\text{WRH}}$, DQMLU**	**$\overline{\text{WRL}}$, DQMLL**
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0	1st time at 0	—	Data 7 to 0	—	Assert
	2nd time at 1	—	Data 15 to 8	—	Assert
Word access at 2	1st time at 2	—	Data 7 to 0	—	Assert
	2nd time at 3	—	Data 15 to 8	—	Assert
Longword access at 0	1st time at 0	—	Data 7 to 0	—	Assert
	2nd time at 2	—	Data 15 to 8	—	Assert
	3rd time at 2	—	Data 23 to 16	—	Assert
	4th time at 3	—	Data 31 to 24	—	Assert

9.5.2 Normal Space Interface

(1) Basic Timing

For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 9.5.8, SRAM Interface with Byte Selection. Figure 9.2 shows the basic timings of normal space access. A no-wait normal access is completed in two cycles. The $\overline{BS}$ signal is asserted for one cycle to indicate the start of a bus cycle.

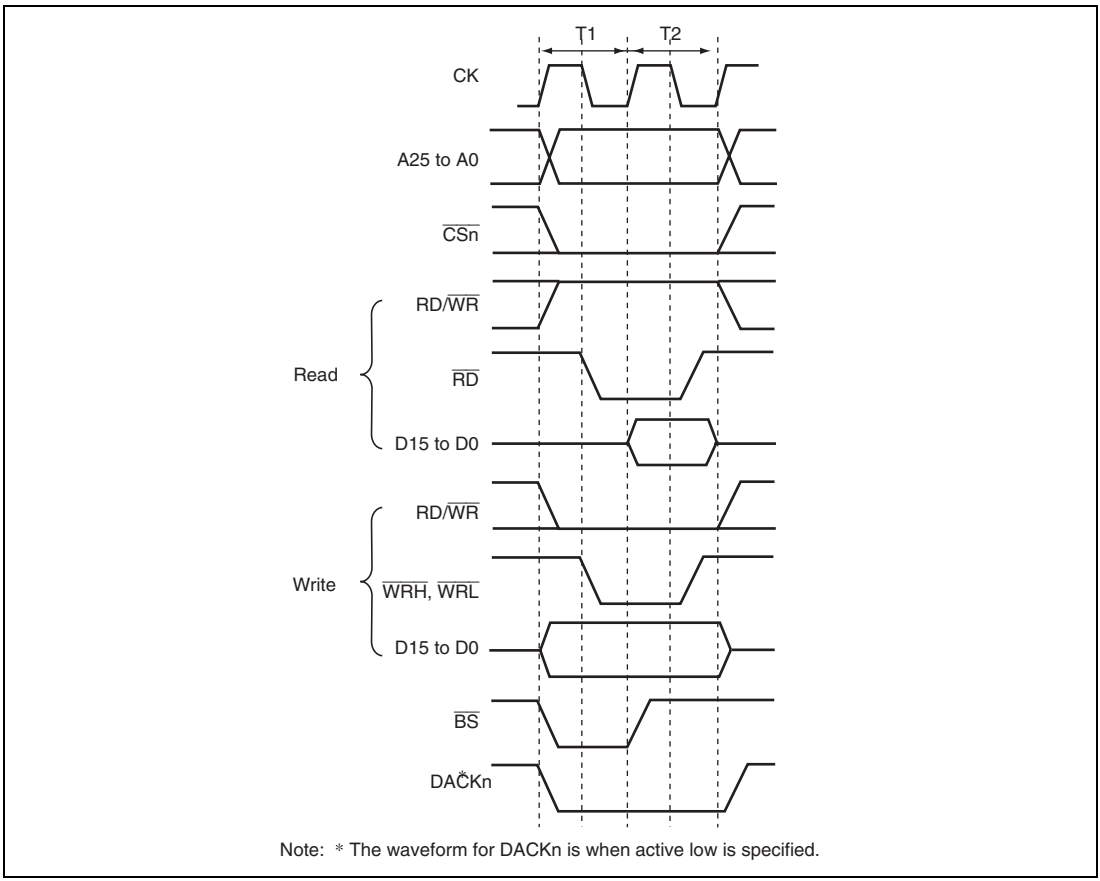


Figure 9.2 Normal Space Basic Access Timing (Access Wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 16 bits are always read in case of a 16-bit device. When writing, only the $\overline{\text{WRxx}}$ signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{\text{RD}}$ when a buffer is established in the data bus. The RD/$\overline{\text{WR}}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 9.3 and 9.4 show the basic timings of normal space access. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to evaluate the external wait (figure 9.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 9.4).

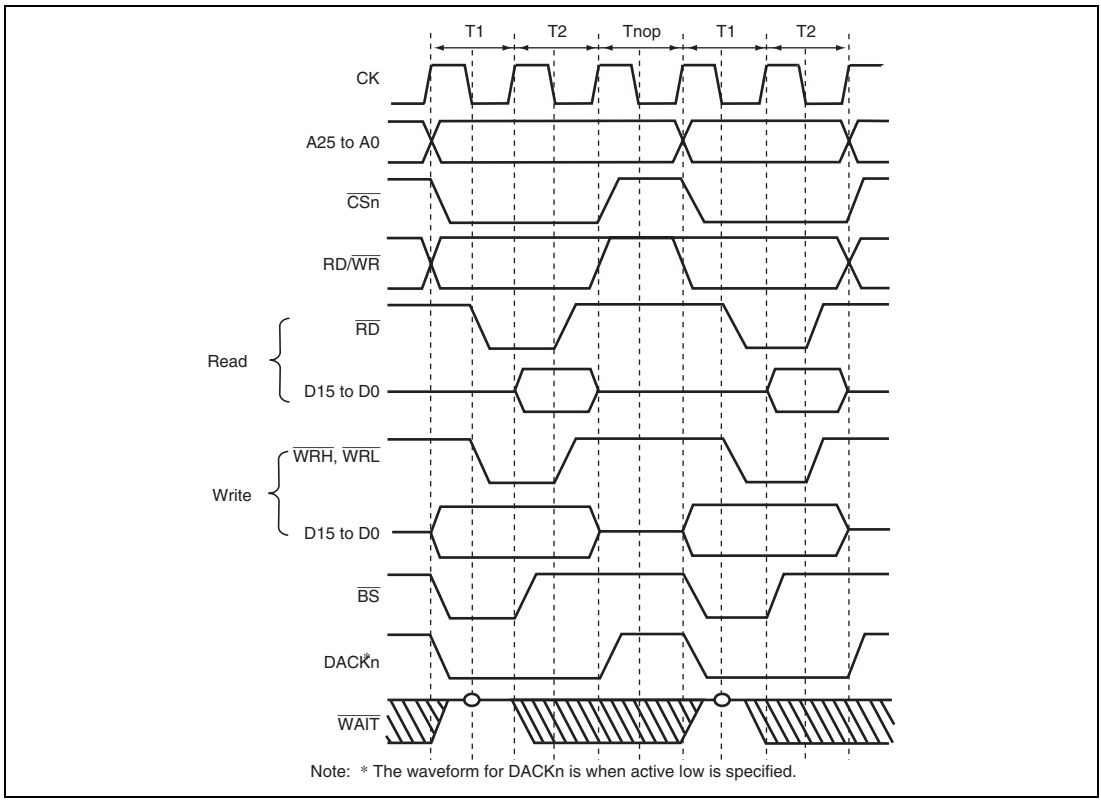


Figure 9.3 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

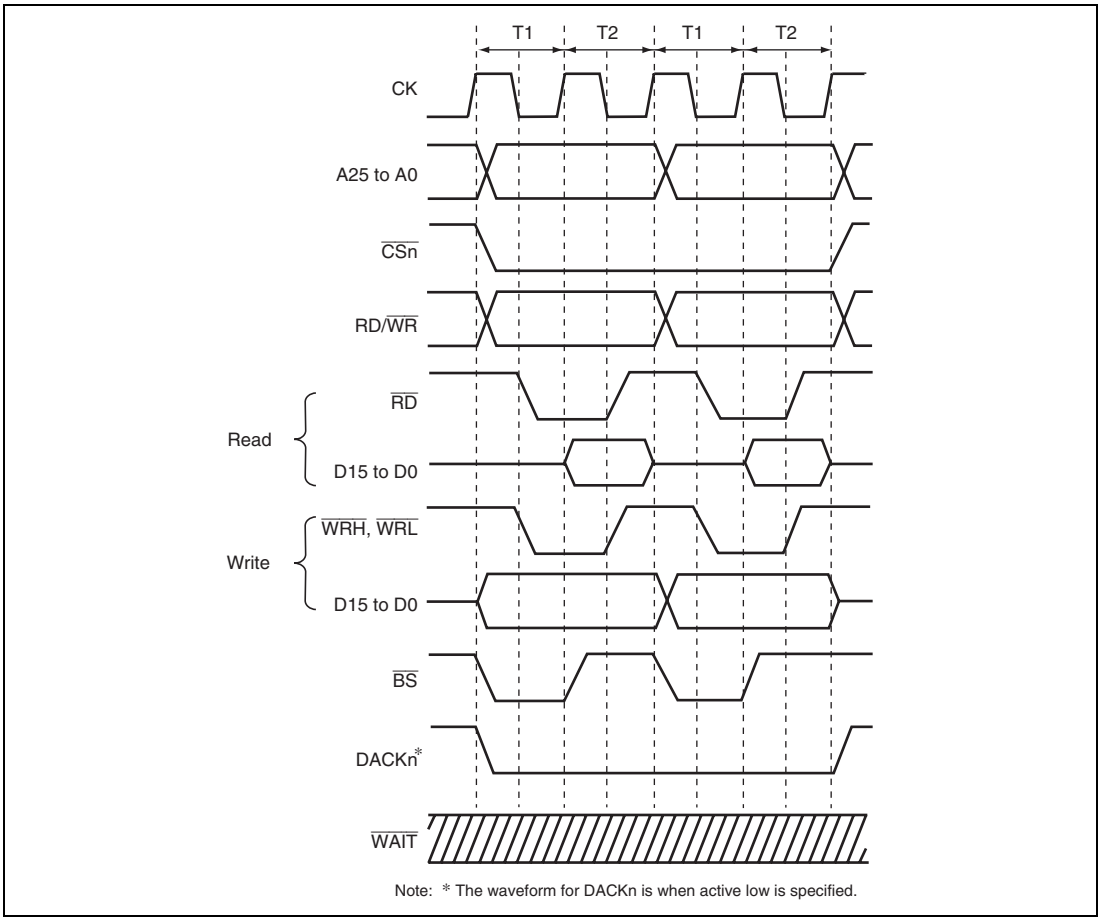


Figure 9.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

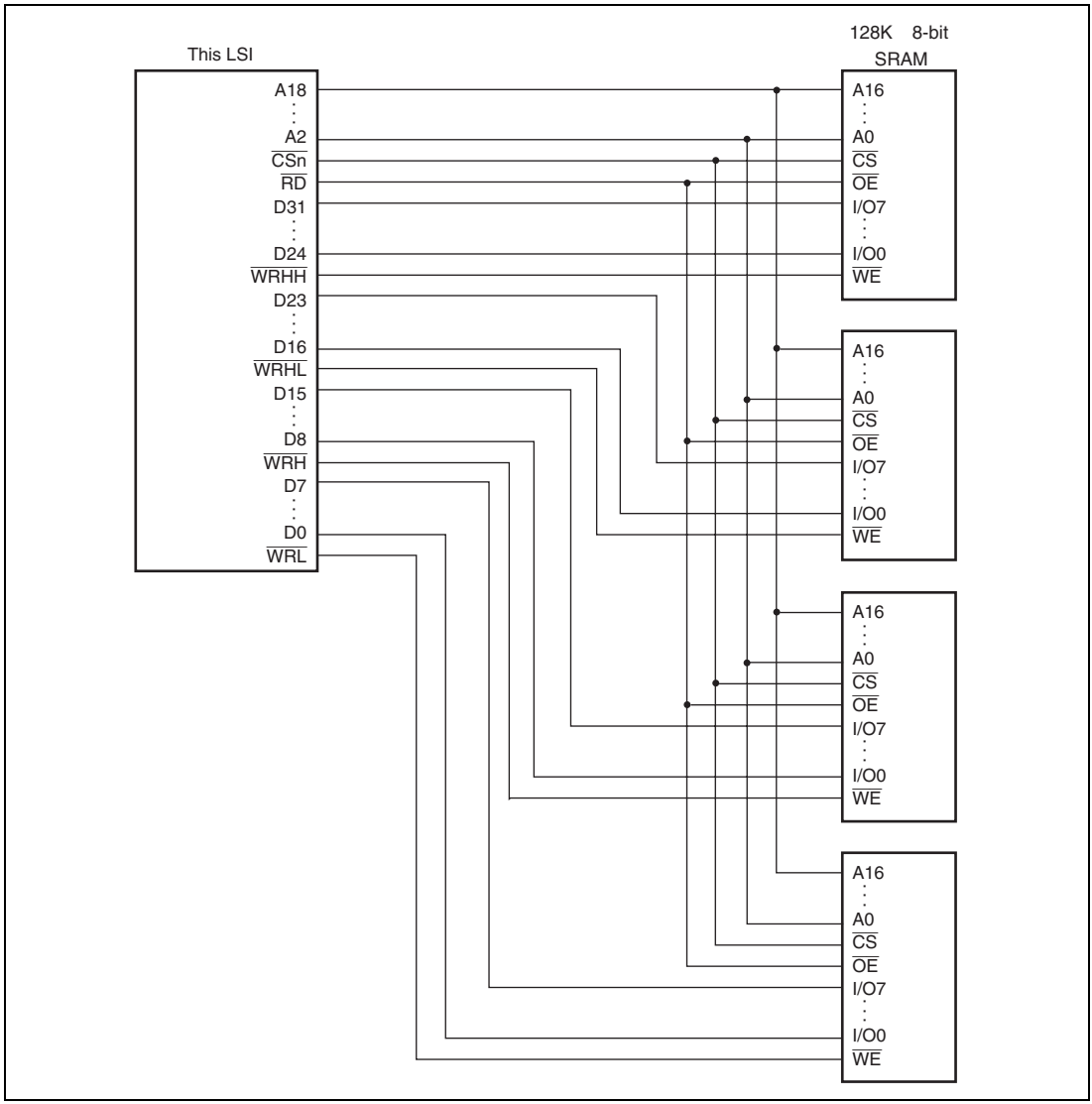


Figure 9.5 Example of 32-Bit Data-Width SRAM Connection

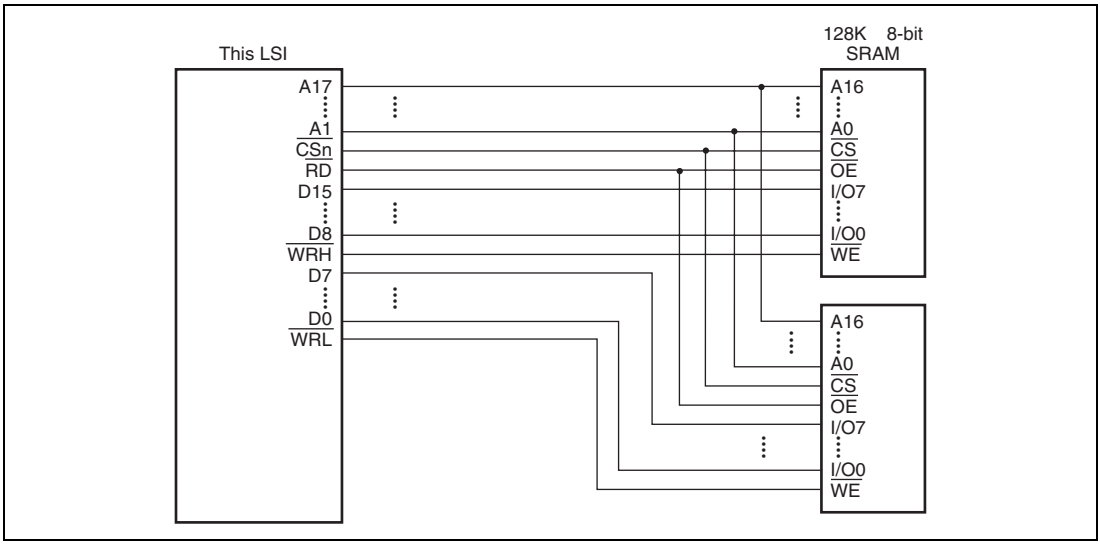


Figure 9.6 Example of 16-Bit Data-Width SRAM Connection

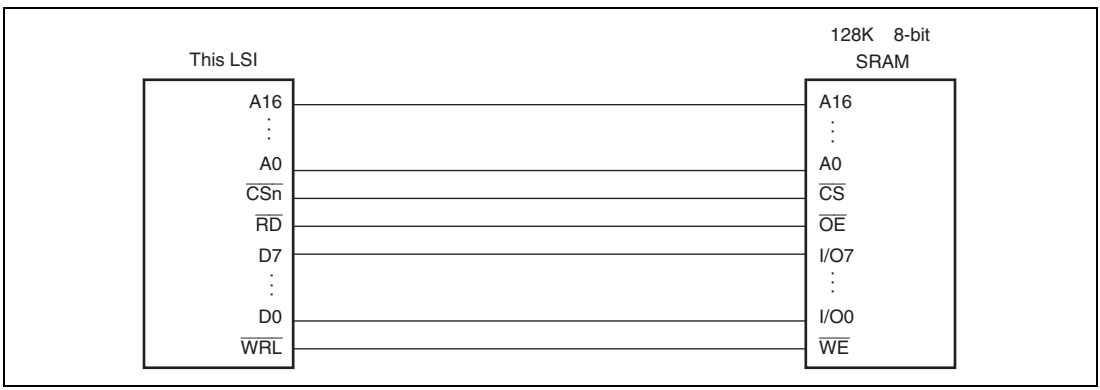


Figure 9.7 Example of 8-Bit Data-Width SRAM Connection

9.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 1, 4, 5, and 7 to insert wait cycles independently in read access and in write access. Areas 0, 2, 3, and 6 have common access wait for read cycle and write cycle. The specified number of Tw cycles are inserted as wait cycles in a normal space access shown in figure 9.8.

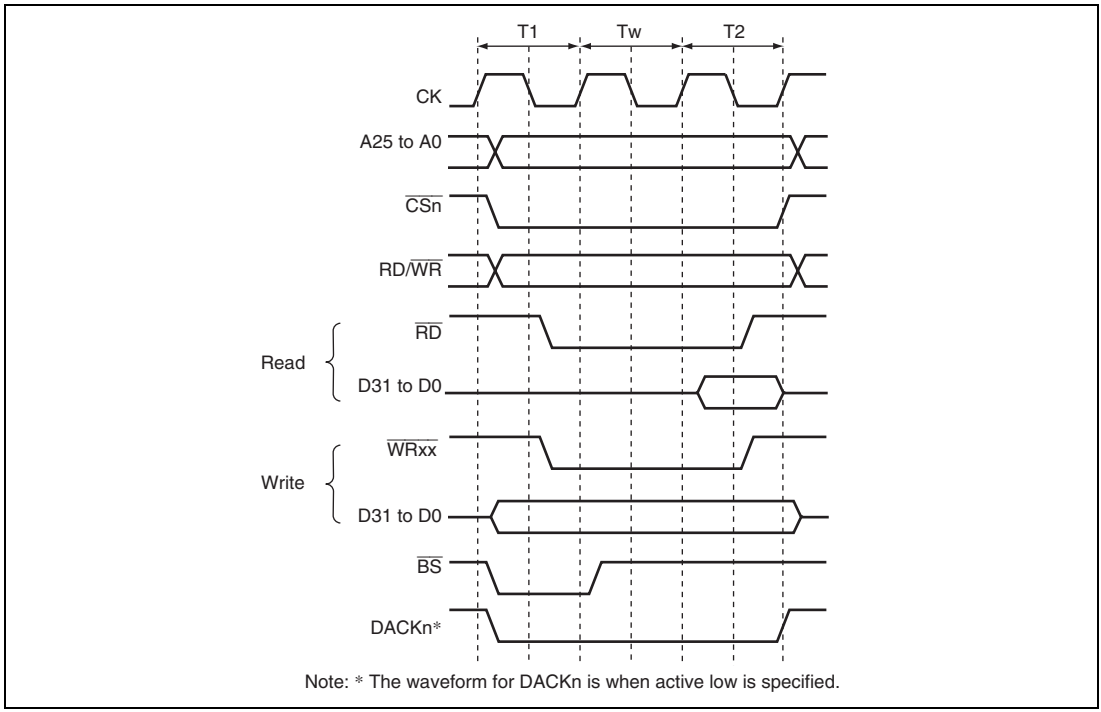


Figure 9.8 Wait Timing for Normal Space Access (Software Wait Only)

When the WM bit in CSnWCR is cleared to 0, the external wait input $\overline{\text{WAIT}}$ signal is also sampled. $\overline{\text{WAIT}}$ pin sampling is shown in figure 9.9. A 2-cycle wait is specified as a software wait. The $\overline{\text{WAIT}}$ signal is sampled on the falling edge of CK at the transition from the T1 or Tw cycle to the T2 cycle.

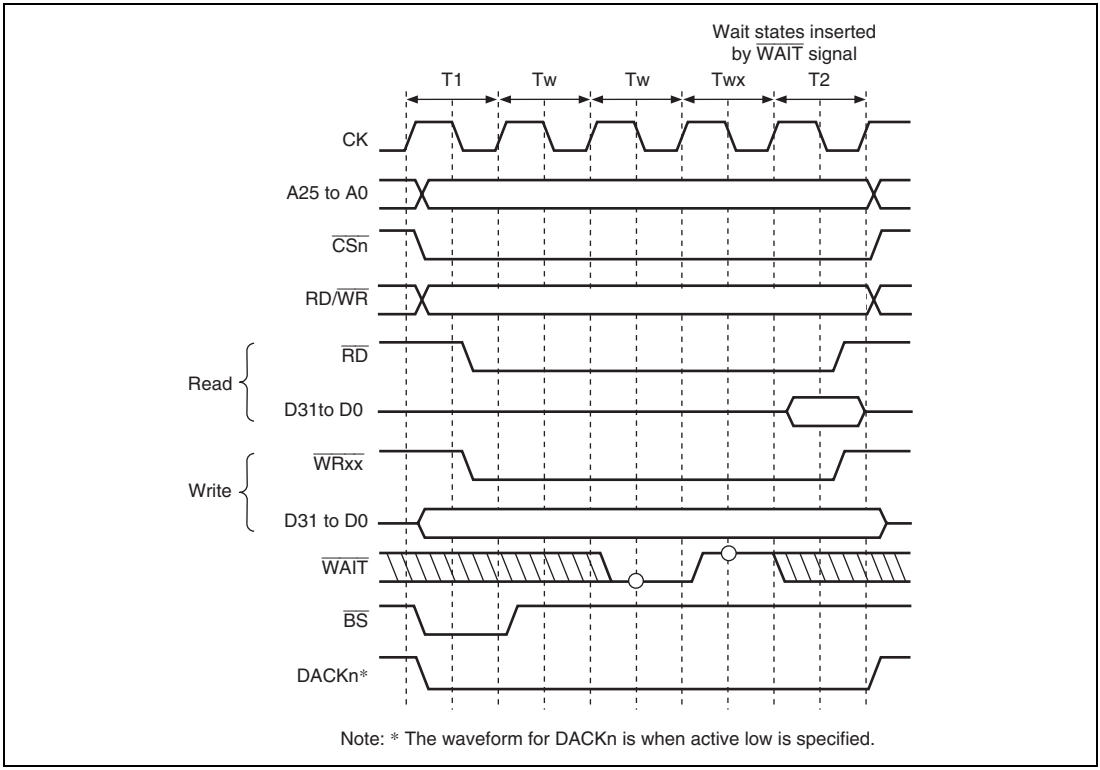


Figure 9.9 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WRxx}}$ negation to $\overline{\text{CSn}}$ negation can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 9.10 shows an example. A Th cycle and a Tf cycle are added before and after an ordinary cycle, respectively. In these cycles, $\overline{\text{RD}}$ and $\overline{\text{WRxx}}$ are not asserted, while other signals are asserted. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

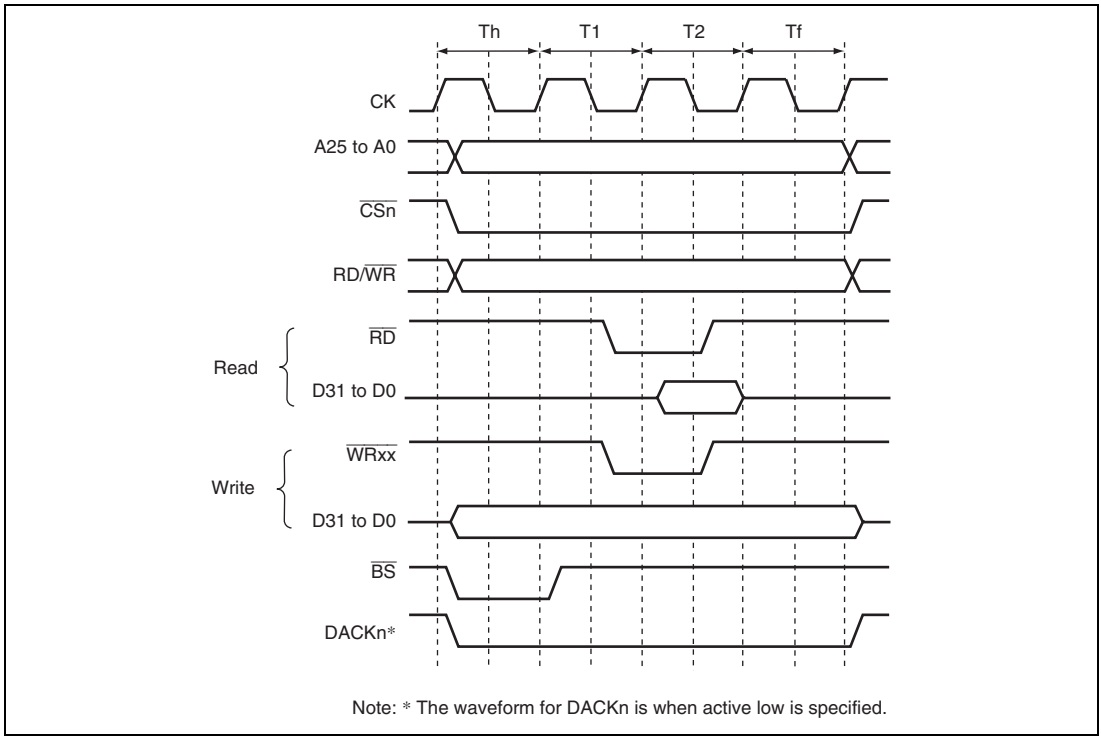


Figure 9.10 $\overline{\text{CSn}}$ Assert Period Expansion

9.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WRxx}$ signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to a normal space. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits. Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in CS5WCR to 1.

The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5}$ signal; it is high in the read cycle and low in the write cycle.

The data cycle is the same as that in a normal space access.

The delay cycles the number of which is specified by SW[1:0] are inserted between cycle Ta3 and cycle T1. The delay cycles the number of which is specified by HW[1:0] are added after cycle T2.

Timing charts are shown in figures 9.11 to 9.14.

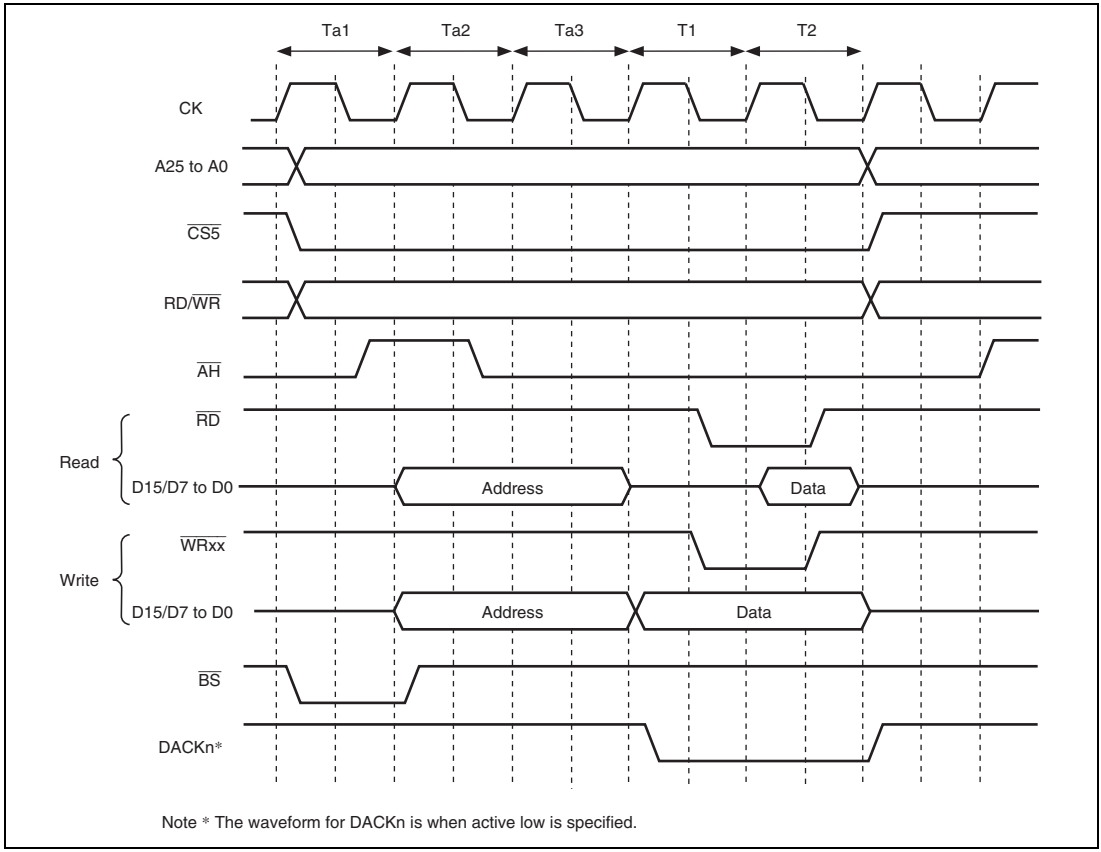


Figure 9.11 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

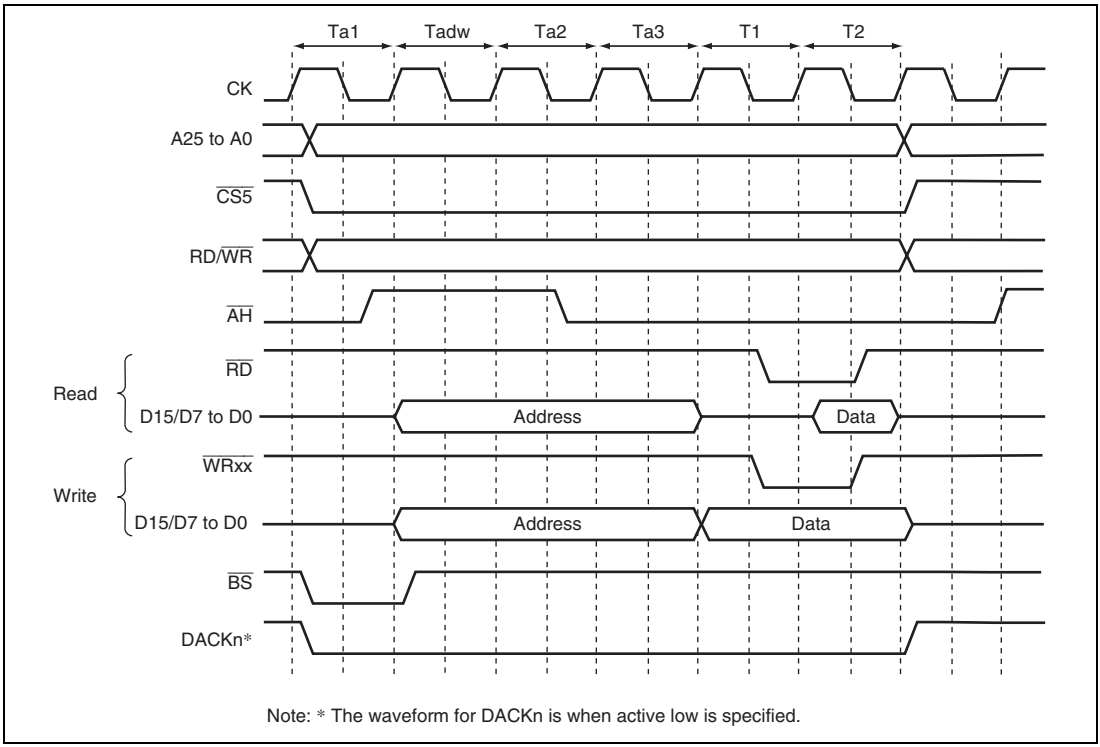


Figure 9.12 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

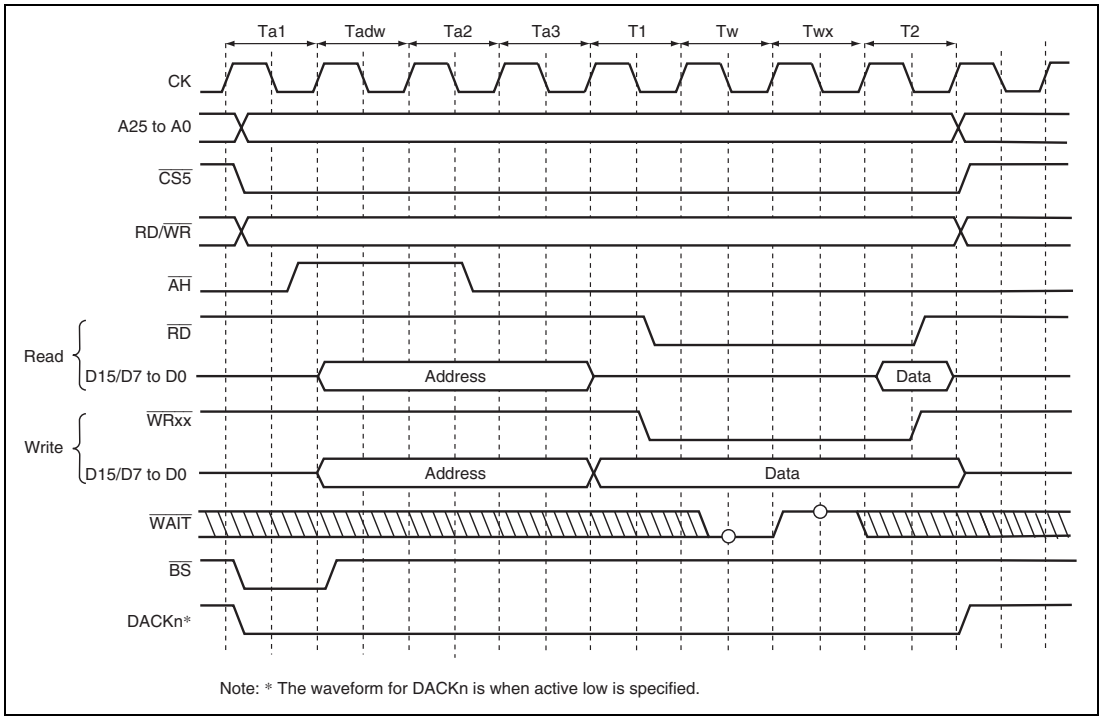


Figure 9.13 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

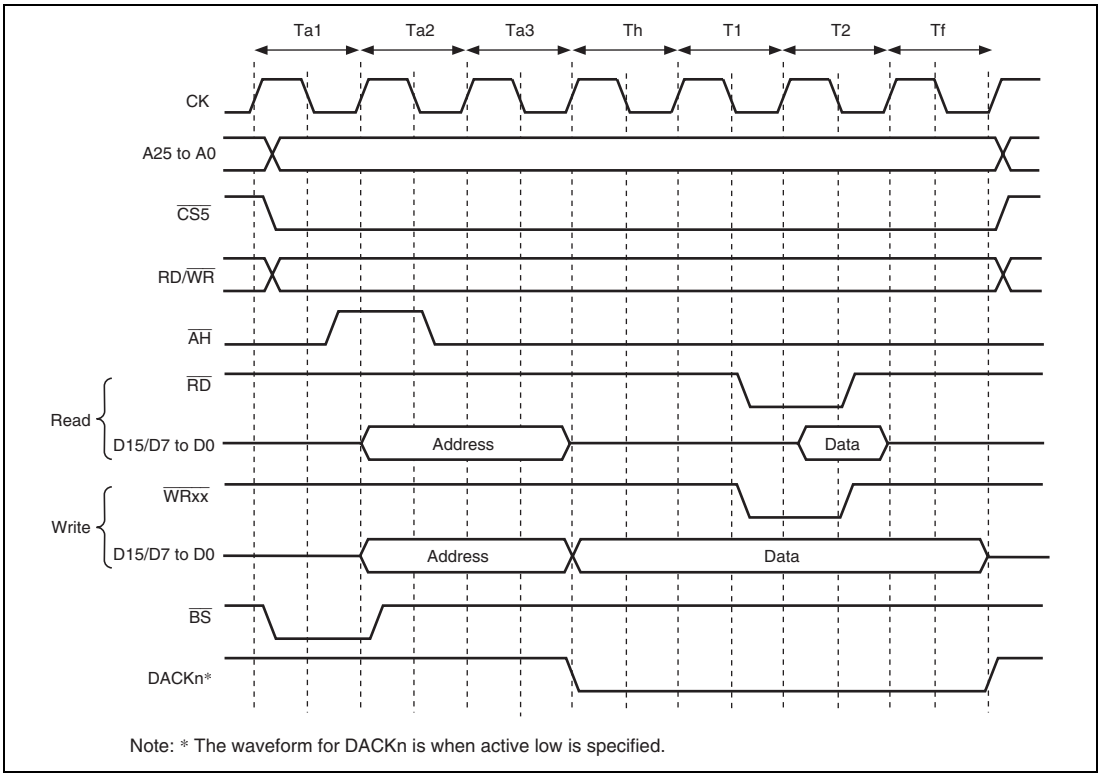


Figure 9.14 Access Timing for MPX Space (Address Cycle No Wait, Assertion Extension Cycle 1.5, Data Cycle No Wait, Negation Extension Cycle 1.5)

9.5.6 SDRAM Interface

(1) SDRAM Direct Connection

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.

The control signals for direct connection of SDRAM are $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, $\overline{\text{CASU}}$, RD/$\overline{\text{WR}}$, DQMUU, DQMUL, DQMLU, DQMLL, CKE, $\overline{\text{CS2}}$, and $\overline{\text{CS3}}$. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to 32 bits or 16 bits.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, and specific address signals. These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS, EMRS)

The byte to be accessed is specified by DQMUU, DQMUL, DQMLU, and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx and the byte to be accessed, see section 9.5.1, Endian/Access Size and Data Alignment.

Figures 9.15 to 9.17 show examples of the connection of the SDRAM with the LSI.

As shown in figure 9.17, two sets of SDRAMs of 32Mbytes or smaller can be connected to the same CS space by using $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, and $\overline{\text{CASL}}$. In this case, a total of 8 banks are assigned to the same CS space: 4 banks specified by $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$, and 4 banks specified by $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$. When accessing the address with A25 = 0, $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$ are asserted. When accessing the address with A25 = 1, $\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are asserted.

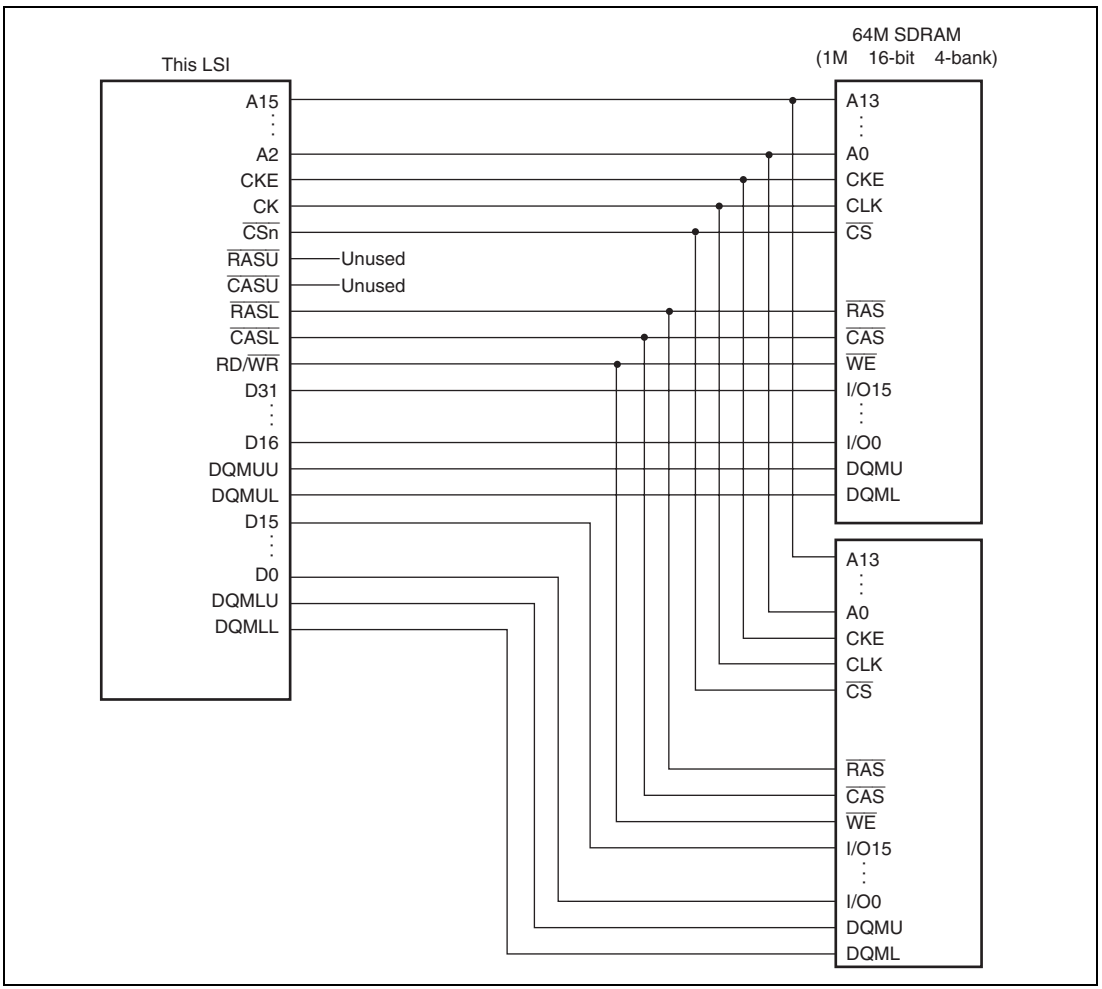


Figure 9.15 Example of 32-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

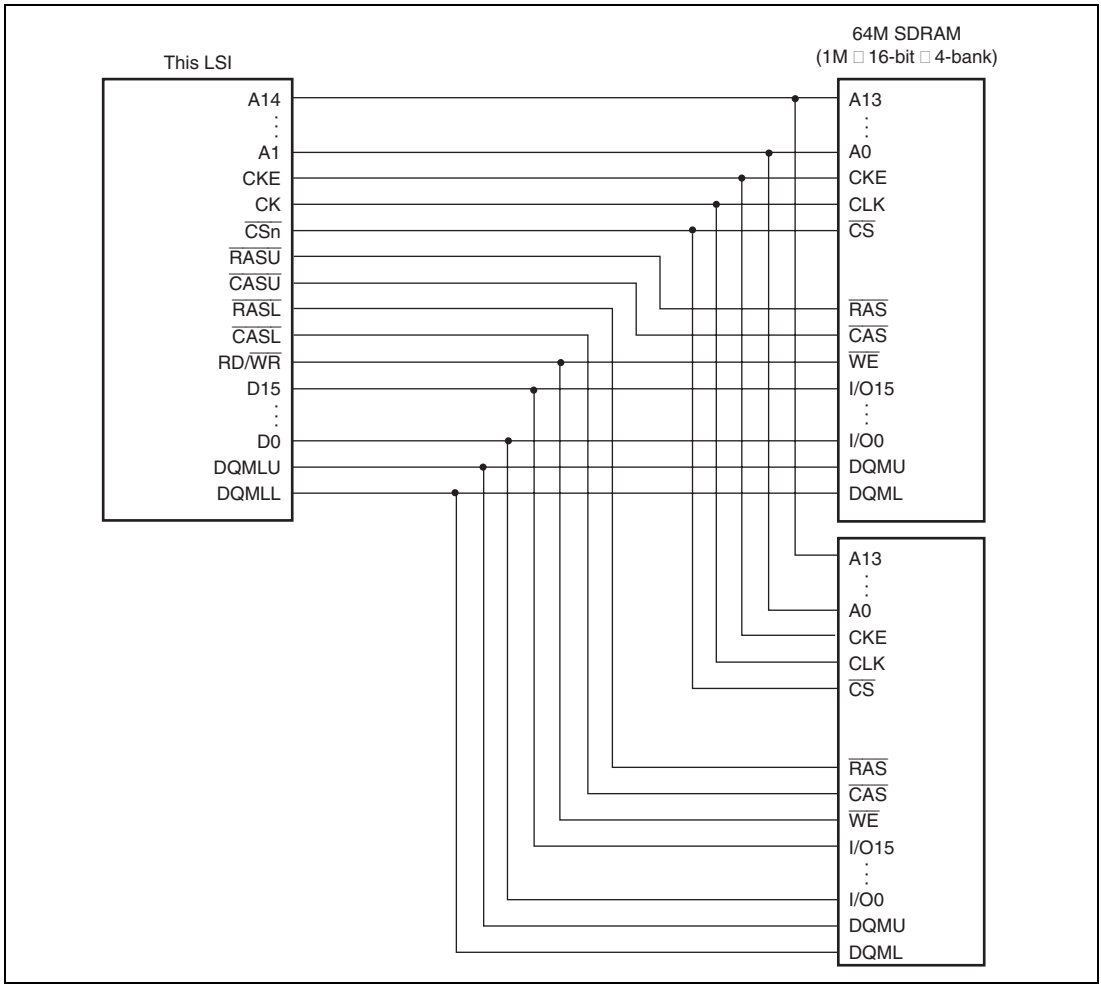


Figure 9.16 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Used)

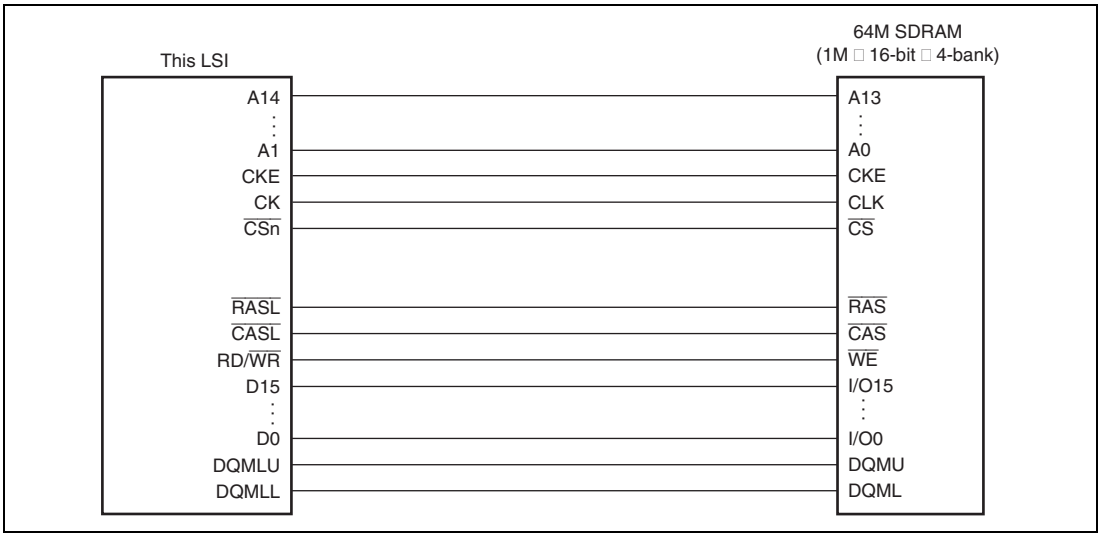


Figure 9.17 Example of 16-Bit Data Width SDRAM Connection

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR, bits A2ROW[1:0], and A2COL[1:0], A3ROW[1:0], and A3COL[1:0] in SDCR. Tables 9.11 to 9.16 show the relationship between the settings of bits BSZ[1:0], A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A29 to A18 are not multiplexed and the original values of address are always output at these pins.

The A0 pin of SDRAM specifies a word address. Therefore, connect the A0 pin of SDRAM to the A1 pin of the LSI; then connect the A1 pin of SDRAM to the A2 pin of the LSI, and so on.

Table 9.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**00 (11 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A12(BA1)	Specifies bank
A13	A21*2	A21*2	A11(BA0)	
A12	A20	L/H*1	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
64-Mbit product (512 Kwords × 32 bits × 4 banks, column 8 bits product): 1				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**01 (12 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A24	A17		Unused
A16	A23	A16		
A15	A23*[2]	A23*[2]	A13(BA1)	Specifies bank
A14	A22*[2]	A22*[2]	A12(BA0)	
A13	A21	A13	A11	Address
A12	A20	L/H*[1]	A10/AP	Specifies address/precharge
A11	A19	A11	A9	Address
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		Unused
A0	A8	A0		
Example of connected memory				
128-Mbit product (1 Mword × 32 bits × 4 banks, column 8 bits product): 1				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**01 (12 Bits)**	**01 (9 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A13(BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A12(BA0)	
A13	A22	A13	A11	Address
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
256-Mbit product (2 Mwords × 32 bits × 4 banks, column 9 bits product): 1				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**01 (12 Bits)**	**10 (10 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*[2]	A25*[2]	A13(BA1)	Specifies bank
A14	A24*[2]	A24*[2]	A12(BA0)	
A13	A23	A13	A11	Address
A12	A22	L/H*[1]	A10/AP	Specifies address/precharge
A11	A21	A11	A9	Address
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		Unused
A0	A10	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 10 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.13 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
11 (32 Bits)	**10 (13 Bits)**	**01 (9 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16	A14(BA1)	Specifies bank
A15	A24*[2]	A25*[2]	A13(BA0)	
A14	A23*[2]	A24*[2]	A12	Address
A13	A22	A13	A11	
A12	A21	L/H*[1]	A10/AP	Specifies address/precharge
A11	A20	A11	A9	Address
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		Unused
A0	A9	A0		
Example of connected memory				
512-Mbit product (4 Mwords × 32 bits × 4 banks, column 9 bits product): 1				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 2				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.14 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 Bits)	**00 (11 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*[2]	A20*[2]	A11 (BA0)	Specifies bank
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.14 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 Bits)	**01 (12 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Specifies bank
A13	A21*2	A21*2	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.15 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 Bits)	**01 (12 Bits)**	**01 (9 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies bank
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.15 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (5)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13 (BA1)	Specifies bank
A13	A23*2	A23*2	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.16 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 9.16 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (6)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2*3	A25*2*3	A14 (BA1)	Specifies bank
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification
3. Only the $\overline{\text{RASL}}$ pin is asserted because the A25 pin specified the bank address. $\overline{\text{RASU}}$ is not asserted.

(3) Burst Read

A burst read occurs in the following cases with this LSI.

- Access size in reading is larger than data bus width.
- 16-byte transfer in DMAC

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 8 times to read 16-byte continuous data from the SDRAM that is connected to a 16-bit data bus. This access is called the burst read with the burst number 8. Table 9.17 shows the relationship between the access size and the number of bursts.

Table 9.17 Relationship between Access Size and Number of Bursts

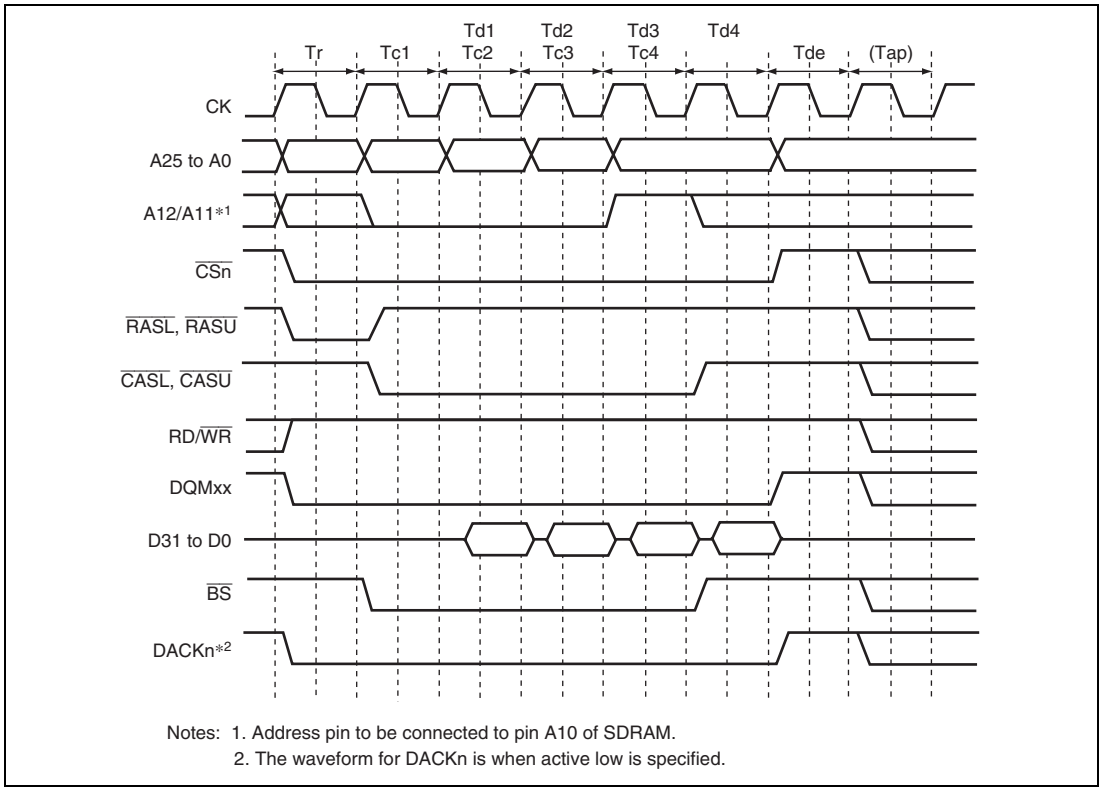
Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
32 bits	8 bits	1
	16 bits	1
	32 bits	1
	16 bytes	4

Figures 9.18 and 9.19 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CK) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.

In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect the SDRAM in variable frequencies. Figure 9.19 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output can be specified using the WTRCD1 and WTRCD0 bits in CS3WCR. If the WTRCD1 and WTRCD0 bits specify one cycles or more, a Trw cycle where the NOT command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles

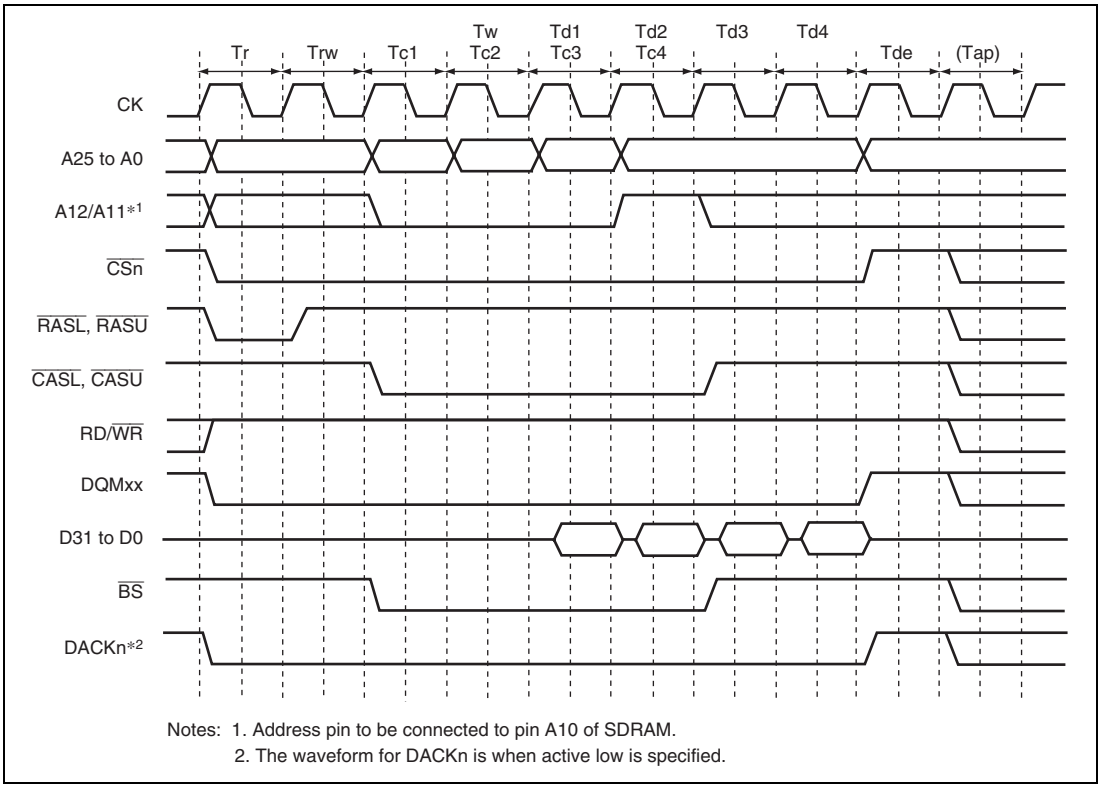
from the Tc1 cycle where the READ command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR and WTRCD0 bit in CS3WCR. The number of cycles from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS latency for the SDRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the SDRAM.

A Tde cycle is an idle cycle required to transfer the read data into this LSI and occurs once for every burst read or every single read.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.18 Burst Read Basic Timing (CAS Latency 1, Auto-Precharge)



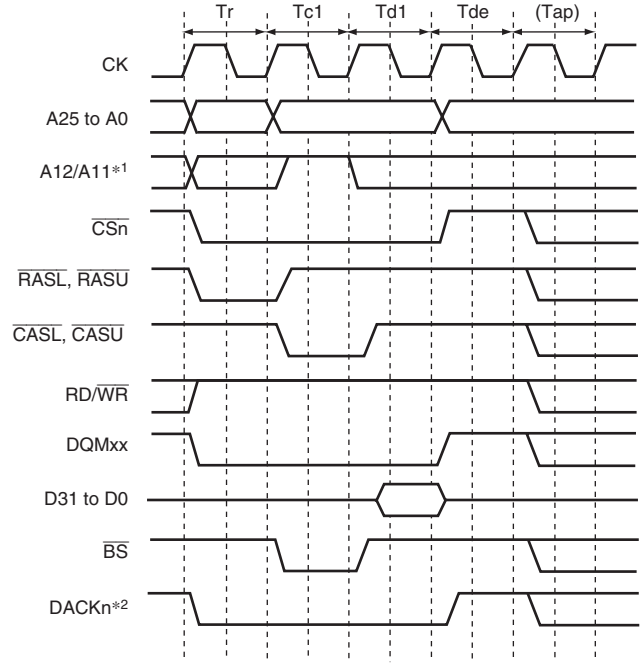
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.19 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD[1:0] = 1 Cycle, Auto-Precharge)

(4) Single Read

A read access ends in one cycle when the data bus width is larger than or equal to the access size. This, simply stated, is single read. As the SDRAM is set to the burst read with the burst length 1, only the required data is output. A read access that ends in one cycle is called single read.

Figure 9.20 shows the single read basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.20 Basic Timing for Single Read (CAS Latency 1, Auto-Precharge)

(5) Burst Write

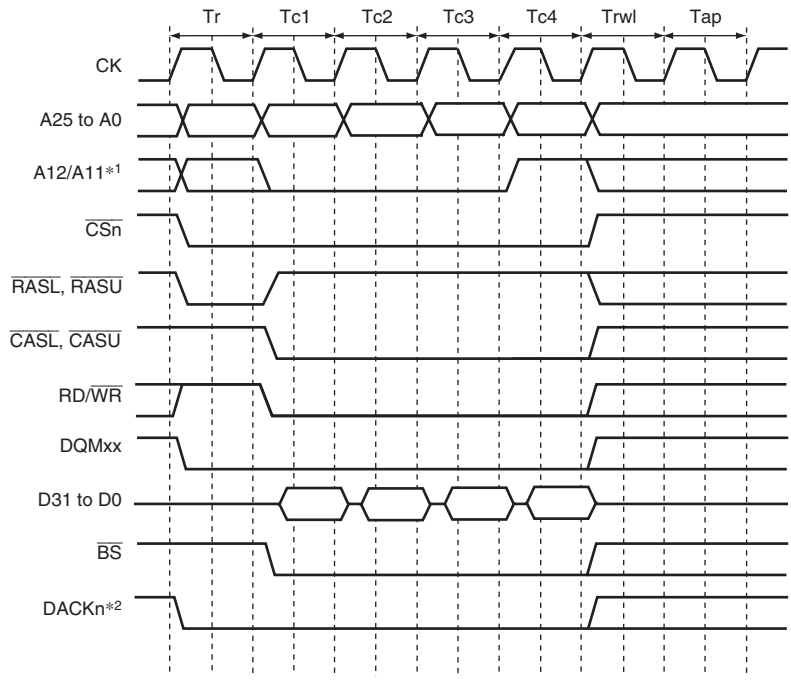
A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- 16-byte transfer in DMAC

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 8 times to write 16-byte continuous data to the SDRAM that is connected to a 16-bit data bus. This access is called burst write with the burst number 8.

The relationship between the access size and the number of bursts is shown in table 9.17.

Figure 9.21 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. Between the Trwl and the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.

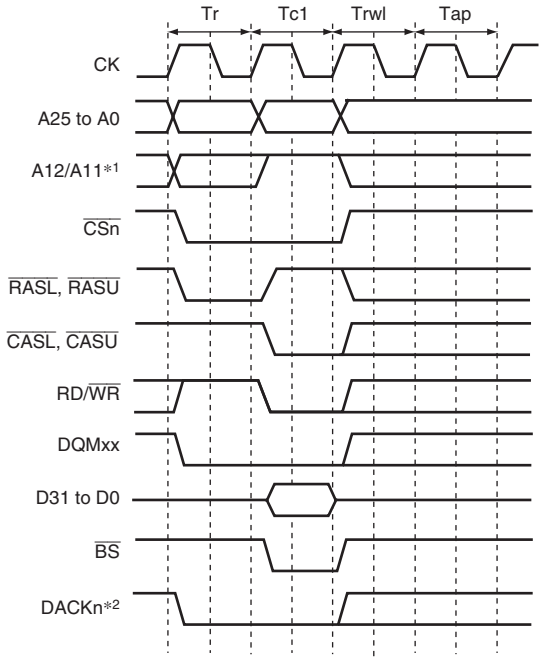


Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.21 Basic Timing for Burst Write (Auto-Precharge)

(6) Single Write

A write access ends in one cycle when the data bus width is larger than or equal to access size. As a single write or burst write with burst length 1 is set in SDRAM, only the required data is output. The write access that ends in one cycle is called single write. Figure 9.22 shows the single write basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.22 Single Write Basic Timing (Auto-Precharge)

(7) Bank Active

The SDRAM bank function can be used to support high-speed access to the same row address. When the BACTV bit in SDCR is 1, access is performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for either the upper or lower bits of area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or SRAM with byte selection. When areas 2 and 3 are both set to SDRAM or both the upper and lower bits of area 3 are connected to SDRAM, auto-precharge mode must be set.

When the bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As SDRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the WTRP1 and WTPR0 bits in CS3WCR.

In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

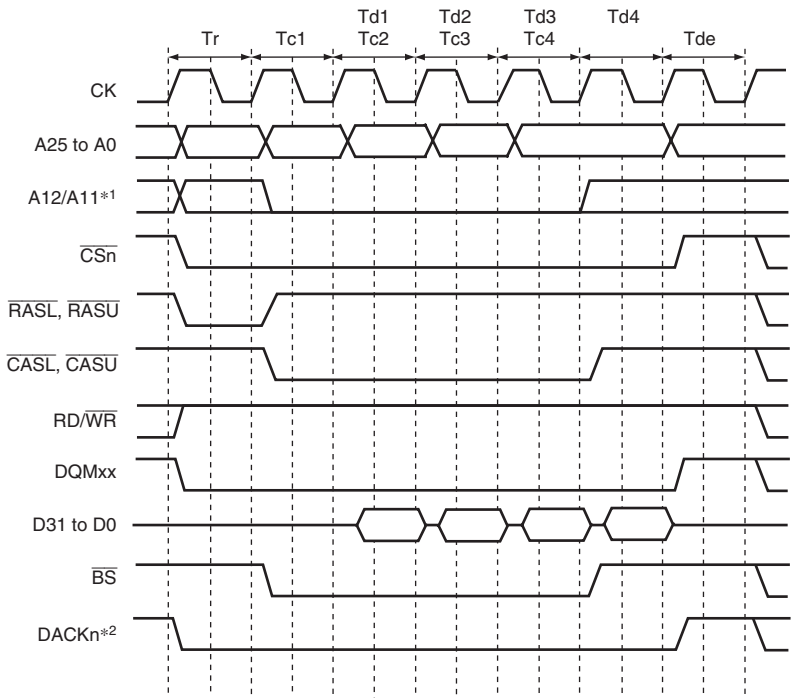
There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 9.23, a burst read cycle for the same row address in figure 9.24, and a burst read cycle for different row addresses in figure 9.25. Similarly, a burst write cycle without auto-precharge is shown in figure 9.26, a burst write cycle for the same row address in figure 9.27, and a burst write cycle for different row addresses in figure 9.28.

In figure 9.24, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS

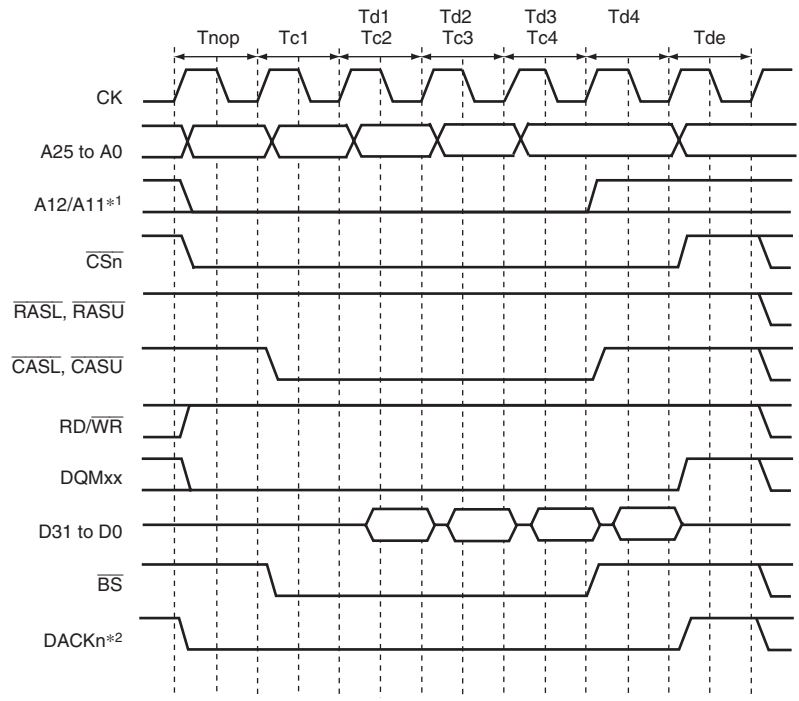
latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the DQMxx signal is asserted after the Tc cycle.

When bank active mode is set, if only access cycles to the respective banks in the area 3 space are considered, as long as access cycles to the same row address continue, the operation starts with the cycle in figure 9.23 or 9.26, followed by repetition of the cycle in figure 9.24 or 9.27. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 9.24 or 9.27 is executed instead of that in figure 9.25 or 9.28. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.



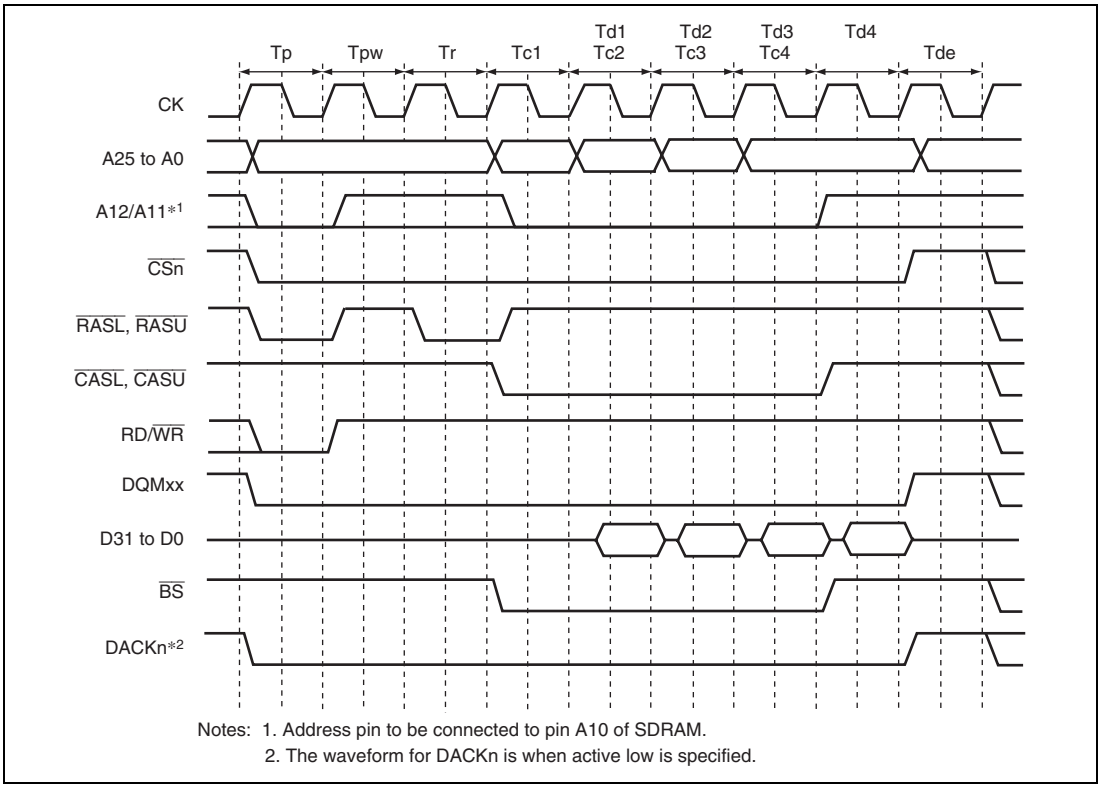
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.23 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)



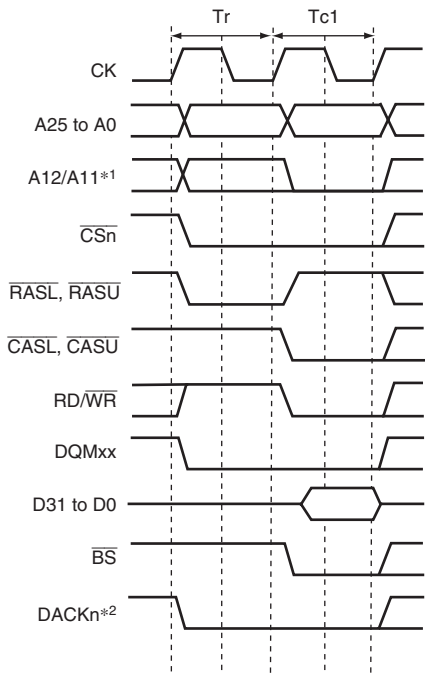
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.24 Burst Read Timing (Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)



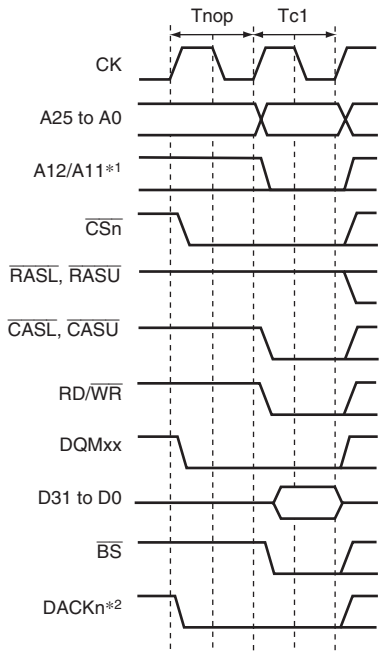
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.25 Burst Read Timing (Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)



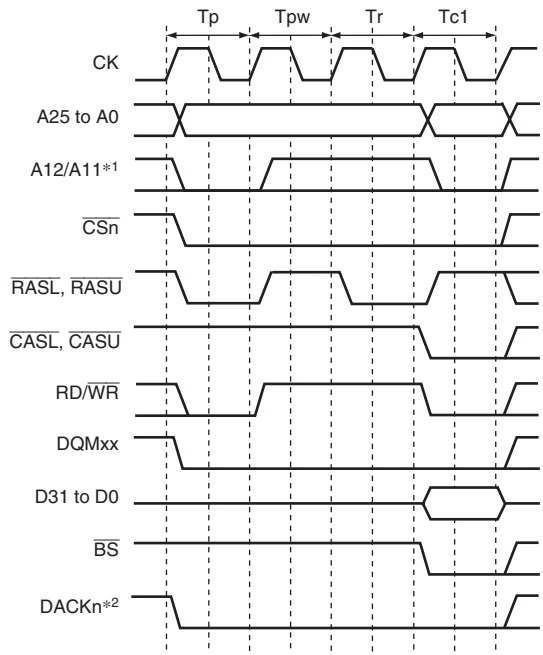
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.26 Single Write Timing (Bank Active, Different Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.27 Single Write Timing (Bank Active, Same Row Addresses in the Same Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.28 Single Write Timing (Bank Active, Different Row Addresses in the Same Bank)

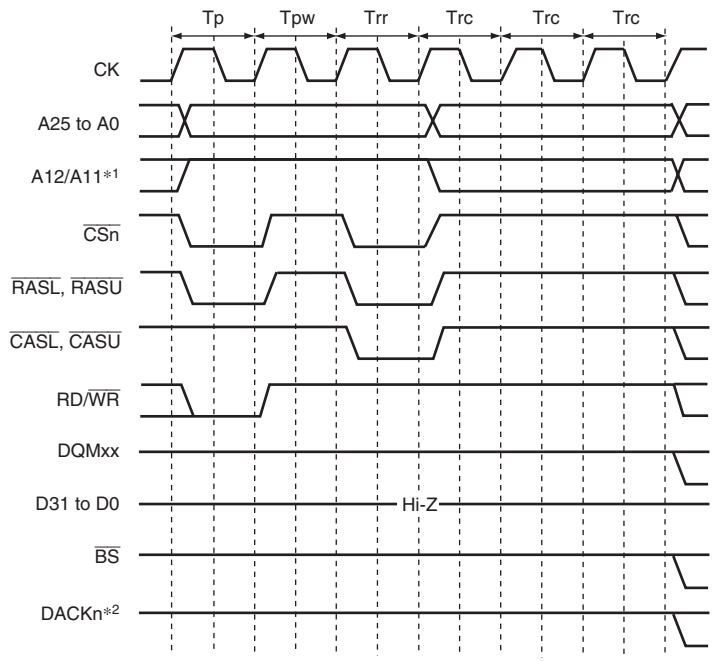
(8) Refreshing

This LSI has a function for controlling SDRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC2 to RRC0 bits in RTCSR. If SDRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

(a) Auto-refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS2 to CKS0 and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC2 to RRC0. At the same time, RTCNT is cleared to zero and the count-up is restarted.

Figure 9.29 shows the auto-refresh cycle timing. After starting, the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to pre-charged state from active state when some bank is being pre-charged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR. A new command is not issued for the duration of the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR after the Trr cycle. The WTRC1 and WTRC0 bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle cycle is inserted between the Tp cycle and Trr cycle when the setting value of the WTRP1 and WTRP0 bits in CS3WCR is longer than or equal to 1 cycle.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.29 Auto-Refresh Timing

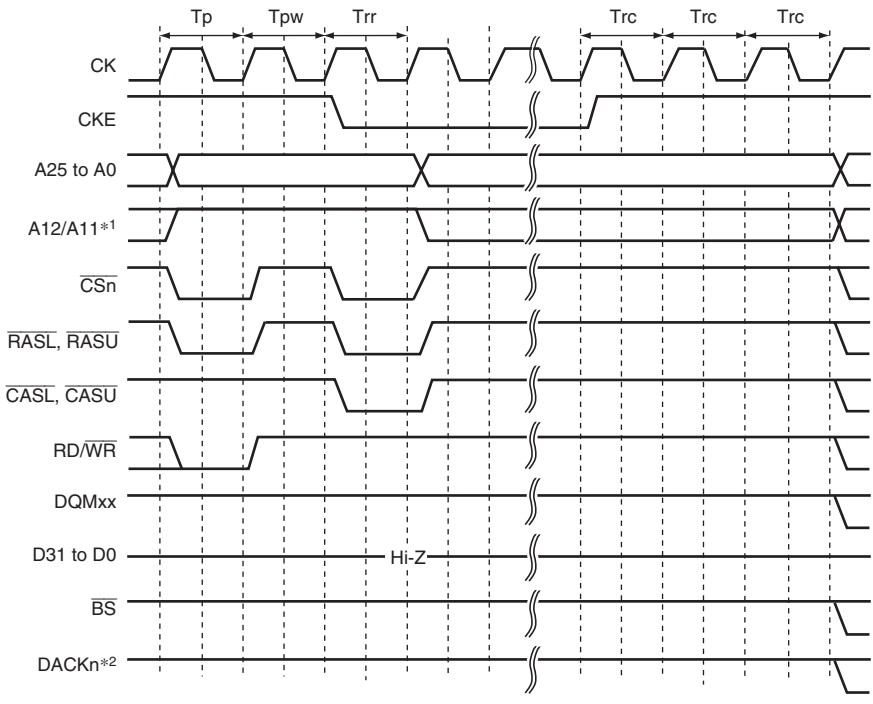
(b) Self-refreshing

Self-refresh mode is a standby mode in which the refresh timing and refresh addresses are generated within the SDRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WSR. SDRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR.

Self-refresh timing is shown in figure 9.30. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode due to an interrupt. Note that the necessary signals such as CKE must be driven even in standby state by setting the HIZCNT bit in CMNCR to 1.

The self-refresh state is not cleared by a manual reset. In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.30 Self-Refresh Timing

(9) Relationship between Refresh Requests and Bus Cycles

If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired. This LSI has the $\overline{\text{REFOUT}}$ pin to request the bus while waiting for refresh execution. For $\overline{\text{REFOUT}}$ pin function selection, see section 22, Pin Function Controller (PFC). This LSI continues to assert $\overline{\text{REFOUT}}$ (low level) until the bus is acquired.

On receiving the asserted $\overline{\text{REFOUT}}$ signal, the external device must negate the $\overline{\text{BREQ}}$ signal and return the bus. If the external bus does not return the bus for a period longer than the specified refresh interval, refresh cannot be executed and the SDRAM contents may be lost.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring.

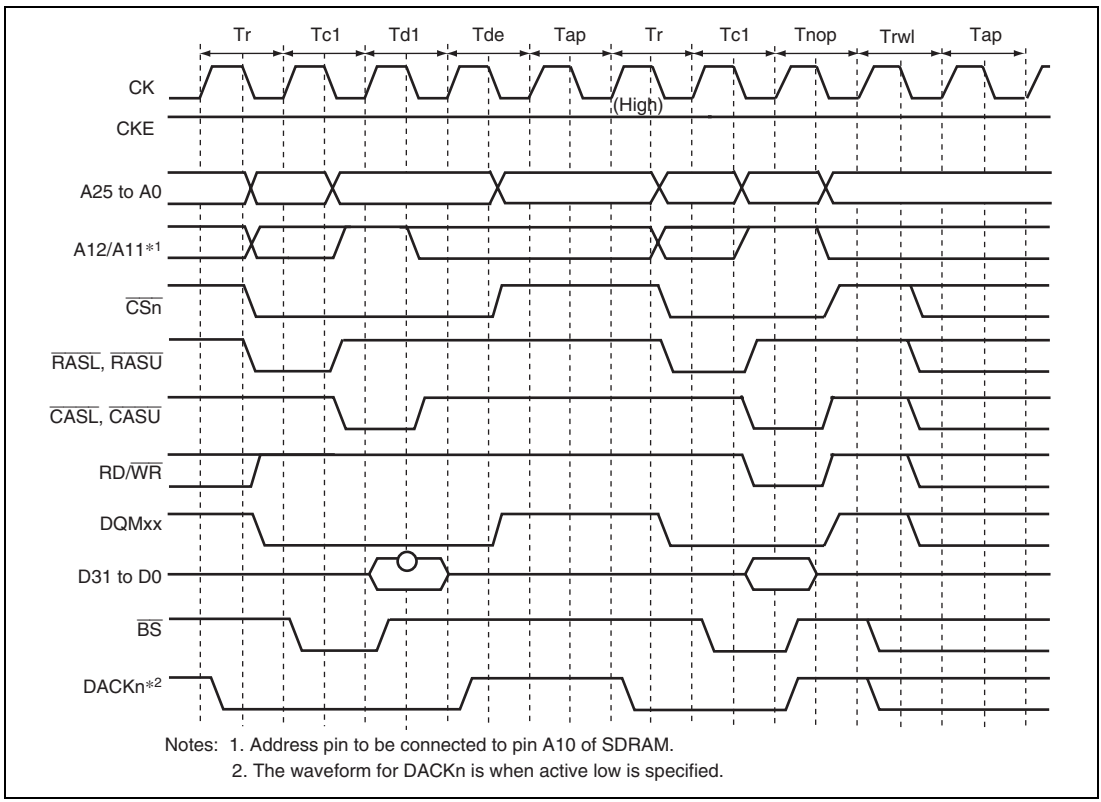
If a bus mastership is requested during self-refresh, the bus will not be released until the refresh is completed.

(10) Low-Frequency Mode

When the SLOW bit in SDCR is set to 1, output of commands, addresses, and write data, and fetch of read data are performed at a timing suitable for operating SDRAM at a low frequency.

Figure 9.31 shows the access timing in low-frequency mode. In this mode, commands, addresses, and write data are output in synchronization with the falling edge of CK, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CK, which is half a cycle faster than the normal timing. This timing allows the hold time of commands, addresses, write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time of commands, addresses, write data, and read data are not guaranteed. Take the operating frequency and timing design into consideration when making the SLOW bit setting.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.31 Low-Frequency Mode Access Timing

(11) Power-Down Mode

If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note that if an access occurs in power-down mode, a cycle of overhead occurs because a cycle is needed to assert the CKE in order to cancel power-down mode.

Figure 9.32 shows the access timing in power-down mode.

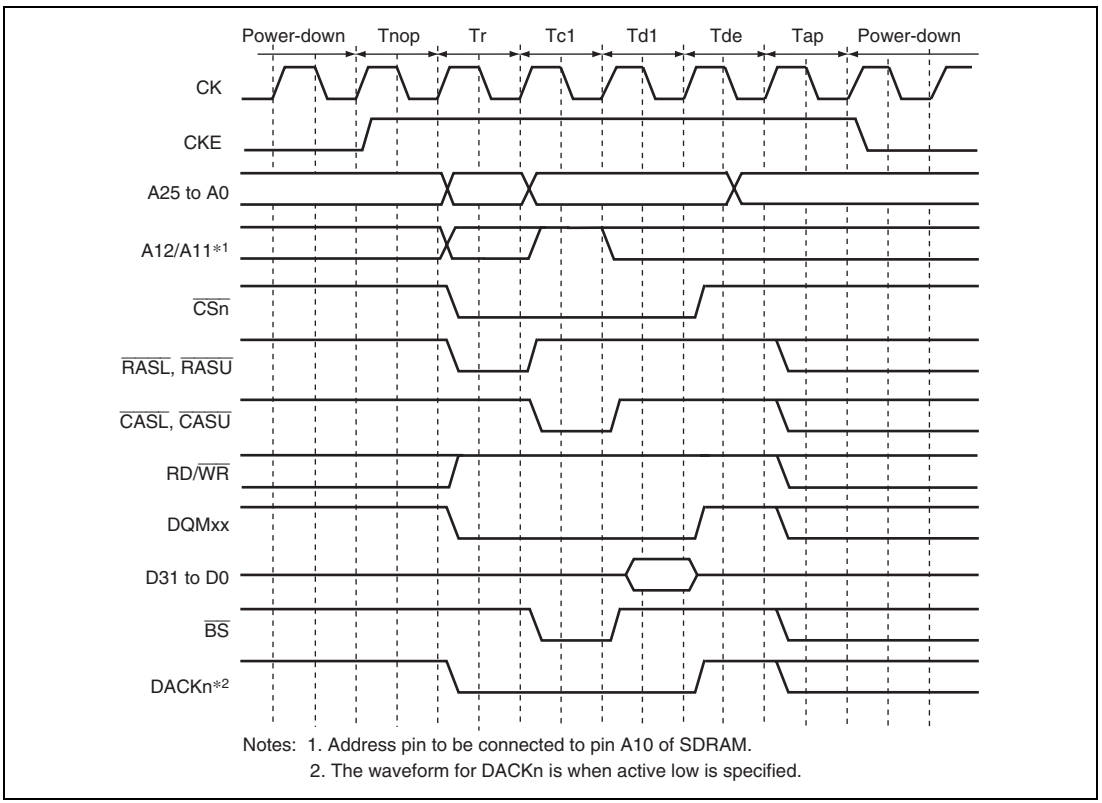


Figure 9.32 Power-Down Mode Access Timing

(12) Power-On Sequence

In order to use SDRAM, mode setting must first be made for SDRAM after waiting for 100 µs or a longer period after powering on. This 100-µs or longer period should be obtained by a power-on reset generating circuit or software.

To perform SDRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the SDRAM mode register. In SDRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{\text{CSn}}$, $\overline{\text{RASU}}$, $\overline{\text{RASL}}$, $\overline{\text{CASU}}$, $\overline{\text{CASL}}$, and RD/$\overline{\text{WR}}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the SDRAM mode register by performing a write to address H'FFFC4000 + X for area 2 SDRAM, and to address H'FFFC5000 + X for area 3 SDRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 2 to 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written in a byte-size access to the addresses shown in table 9.18. In this time 0 is output at the external address pins of A12 or later.

Table 9.18 Access Address in SDRAM Mode Register Write

- Setting for Area 2

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460
32 bits	2	H'FFFC4880	H'0000880
	3	H'FFFC48C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060
32 bits	2	H'FFFC4080	H'0000080
	3	H'FFFC40C0	H'00000C0

- Setting for Area 3

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460
32 bits	2	H'FFFC5880	H'0000880
	3	H'FFFC58C0	H'00008C0

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060
32 bits	2	H'FFFC5080	H'0000080
	3	H'FFFC50C0	H'00000C0

  When a mode register write command is issued, the outputs of the external address pins are as follows.

When the data bus width of the area connected to SDRAM is 32 bits	A15 to A9	00000000 (burst read/burst write) 00000100 (burst read/single write)
	A8 to A6	010 (CAS latency 2), 011 (CAS latency 3)
	A5	0 (lap time = sequential)
	A4 to A2	000 (burst length 1)
When the data bus width of the area connected to SDRAM is 16 bits	A14 to A8	00000000 (burst read/burst write) 00000100 (burst read/single write)
	A7 to A5	010 (CAS latency 2), 011 (CAS latency 3)
	A4	0 (lap time = sequential)
	A3 to A1	000 (burst length 1)

Mode register setting timing is shown in figure 9.33. A PALL command (all bank pre-charge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the WTRC1 and WTRC0 bits in CS3WCR,

are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or more, are inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer to the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer than the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.

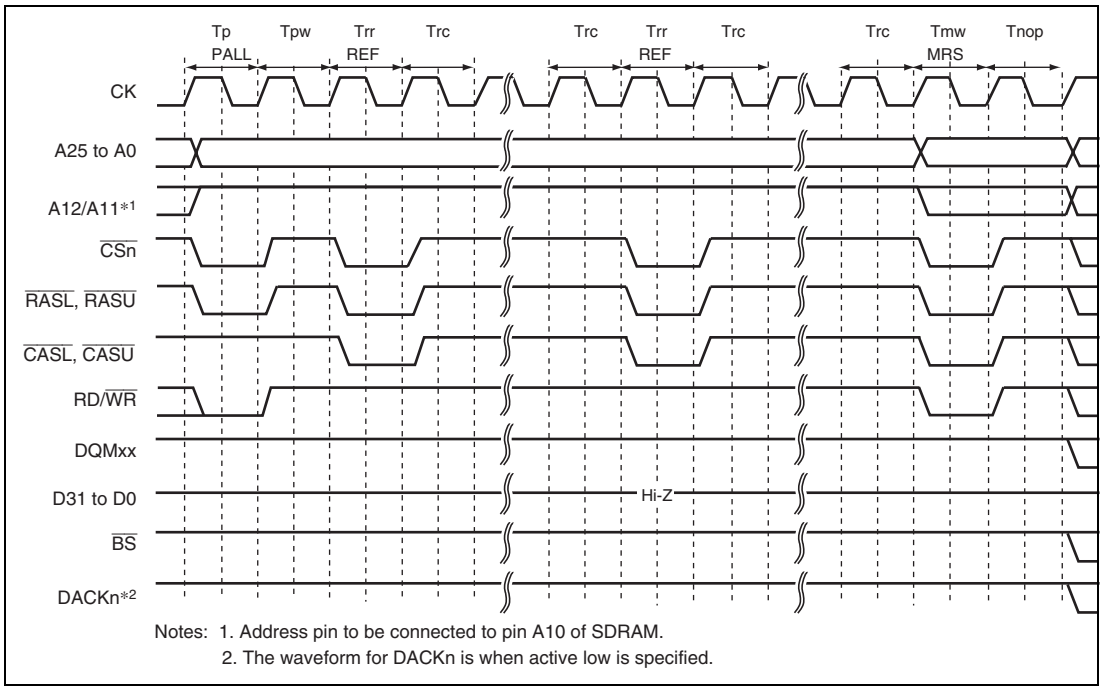


Figure 9.33 SDRAM Mode Write Timing (Based on JEDEC)

(13) Low-Power SDRAM

The low-power SDRAM can be accessed using the same protocol as the normal SDRAM.

The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which there is data in a work area other than the specific area can be lost without severe repercussions.

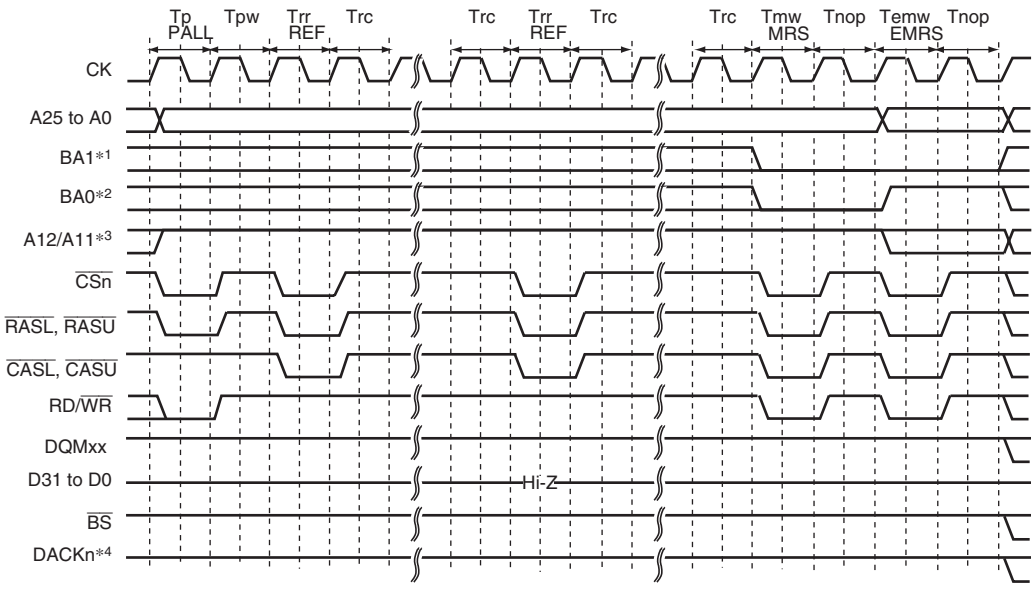
The low-power SDRAM supports the extension mode register (EMRS) in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table below. For example, if data H'0YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XX0 and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

However, since addresses written to this LSI are output without change, set data in accord with the EMRS specifications for the given SDRAM area.

Table 9.19 Output Addresses when EMRS Command Is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'FFFC4XX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'FFFC5XX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'FFFC4XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (with refresh)	H'FFFC5XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS2 MRS + EMRS (without refresh)	H'FFFC4XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (without refresh)	H'FFFC5XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY



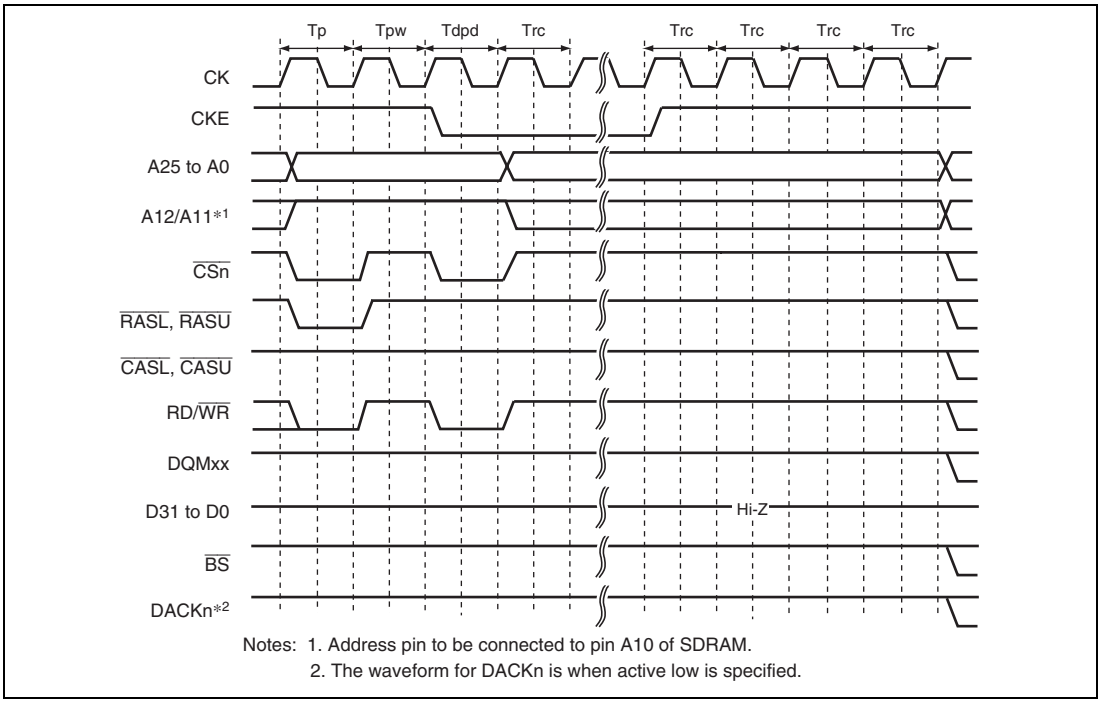
Notes: 1. Address pin to be connected to pin BA1 of SDRAM.
2. Address pin to be connected to pin BA0 of SDRAM.
3. Address pin to be connected to pin A10 of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 9.34 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit in the SDCR is set to 1 while the DEEP and RFSH bits in the SDCR are set to 1, the low-power SDRAM enters deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel deep power-down mode. Before executing an access after returning from deep power-down mode, the power-up sequence must be re-executed.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 9.35 Deep Power-Down Mode Transition Timing

9.5.7 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called burst mode or page mode. In a burst ROM (clock asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent access cycles are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent access cycles, addresses are changed at the falling edge of the CK.

For the 1st access cycle, the number of wait cycles specified by the W3 to W0 bits in CSnWCR is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the W1 to W0 bits in CSnWCR is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that does not perform the burst operation in the burst ROM (clock asynchronous) interface, access timing is same as a normal space.

Table 9.20 lists a relationship between bus width, access size, and the number of bursts. Figure 9.36 shows a timing chart.

Table 9.20 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes*2	x0	16	1
		10	4	4
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes*2	00	8	1
		01	2	4
		10*1	4	2
			2, 4, 2	3
32 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	1	1
	16 bytes*2	Not affected	4	1

Notes: 1. When the bus width is 16 bits, the access size is 16 bits, and the BST[1:0] bits in CSnWCR are 10, the number of bursts and access count depend on the access start address. At address H'xxx0 or H'xxx8, 4-4 burst access is performed. At address H'xxx4 or H'xxxC, 2-4-2 burst access is performed.

2. Only the DMAC is capable of transfer with 16 bytes as the unit of access. The maximum unit of access for the DTC, E-DMAC, and CPU is 32 bits.

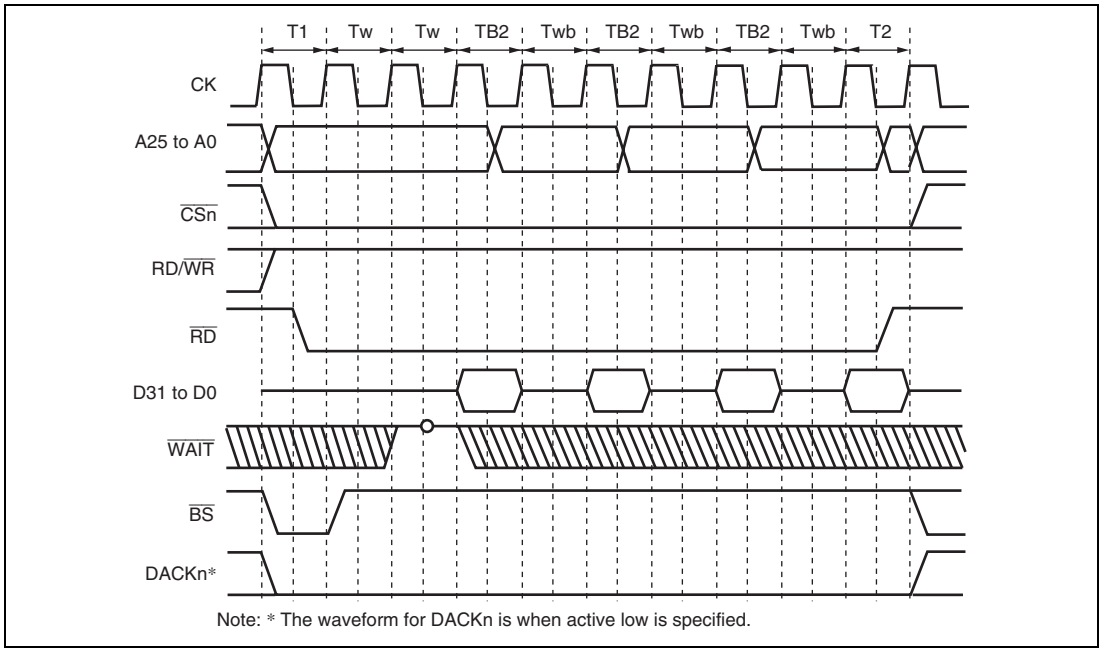


Figure 9.36 Burst ROM Access Timing (Clock Asynchronous)
(Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

9.5.8 SRAM Interface with Byte Selection

The SRAM interface with byte selection is for access to an SRAM which has a byte-selection pin (WRxx). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the SRAM interface with byte selection is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the WRxx pin, which is different from that for the normal space interface. The basic access timing is shown in figure 9.37. In write access, data is written to the memory according to the timing of the byte-selection pin (WRxx). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the WRxx pin and RD/WR pin timings change. Figure 9.38 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/WR). The data hold timing from RD/WR negation to data write must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 9.39 shows the access timing when a software wait is specified.

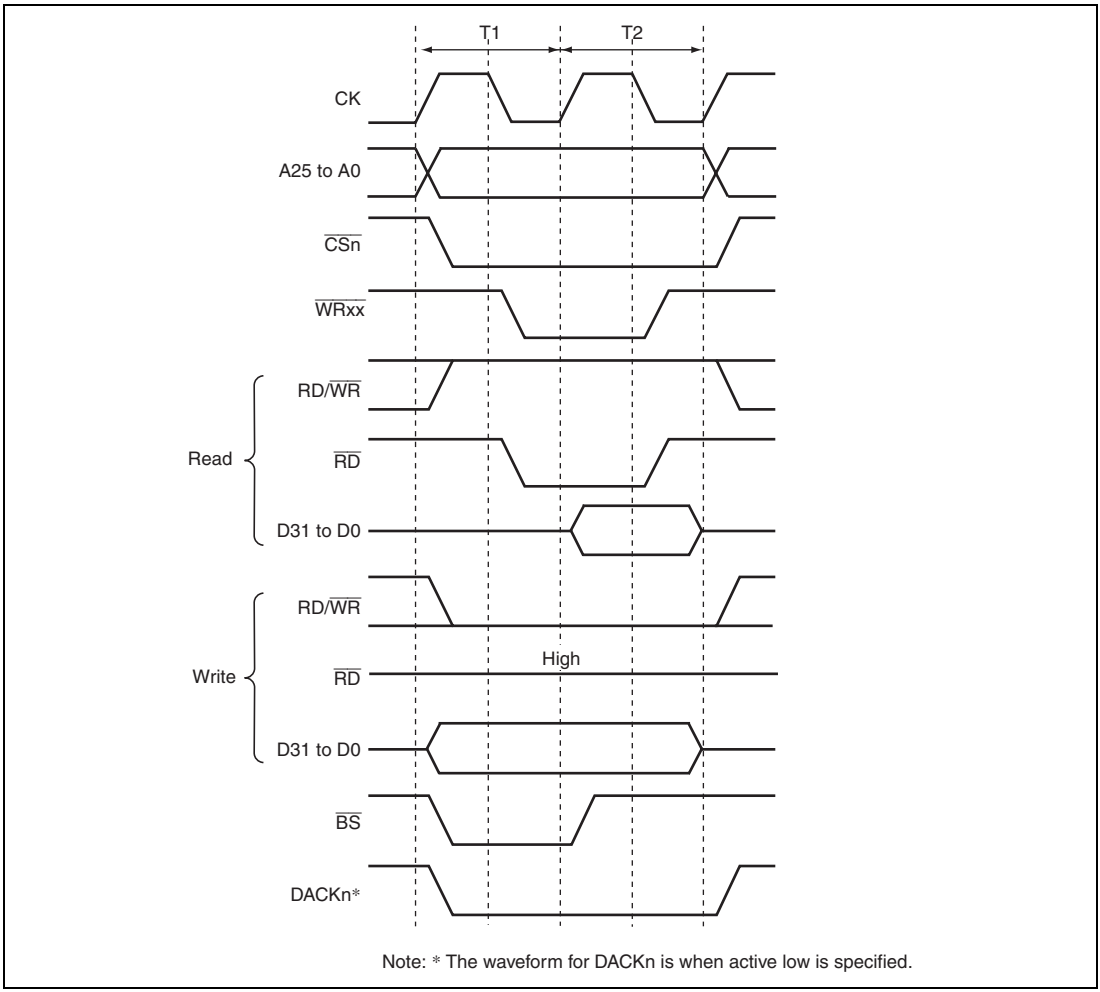


Figure 9.37 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

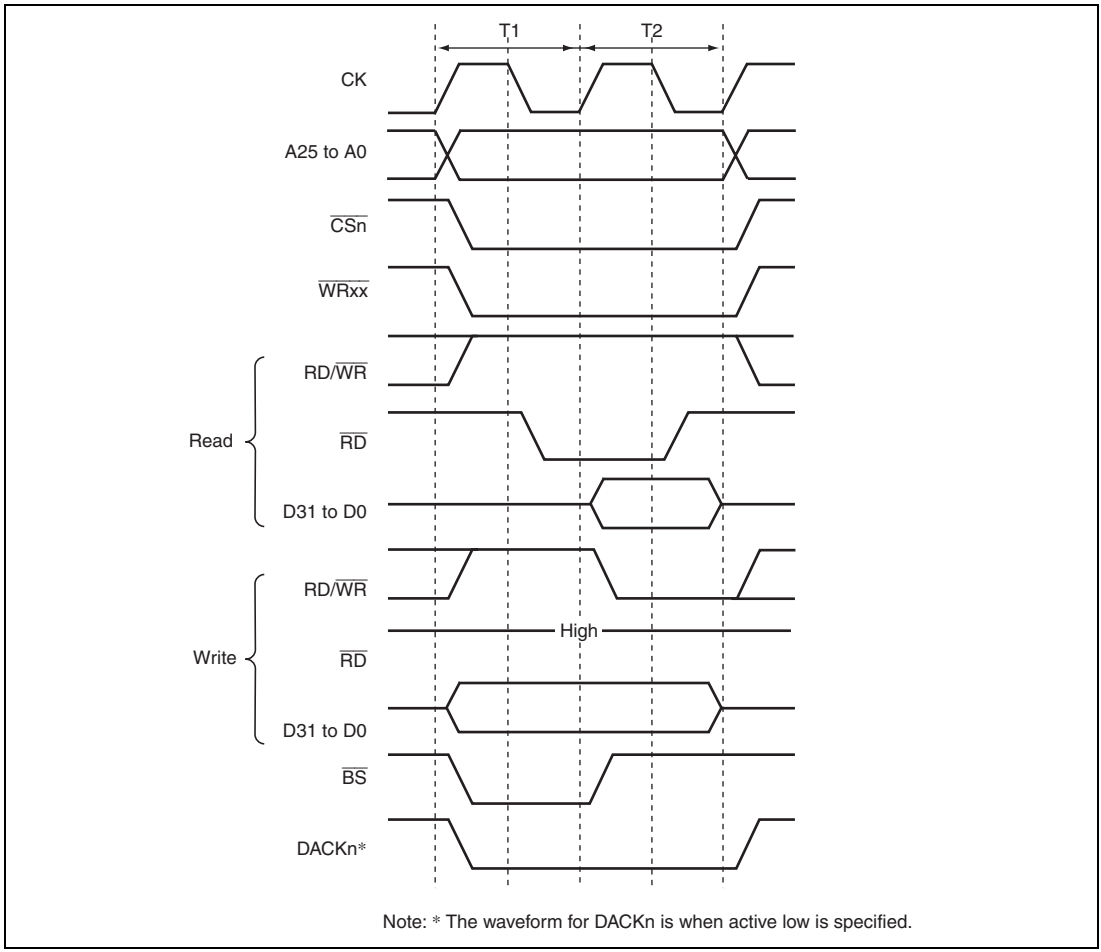


Figure 9.38 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

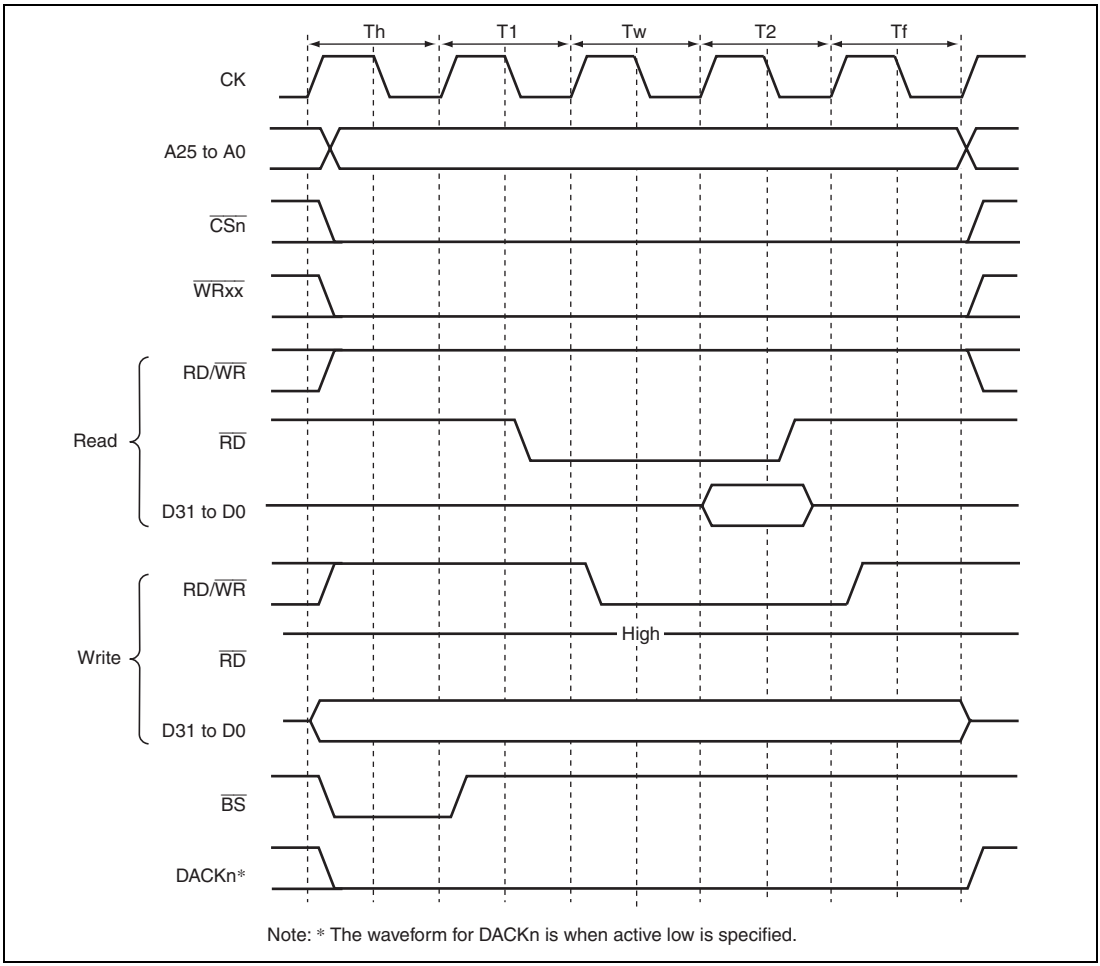


Figure 9.39 Wait Timing for SRAM with Byte Selection (BAS = 1)
(SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

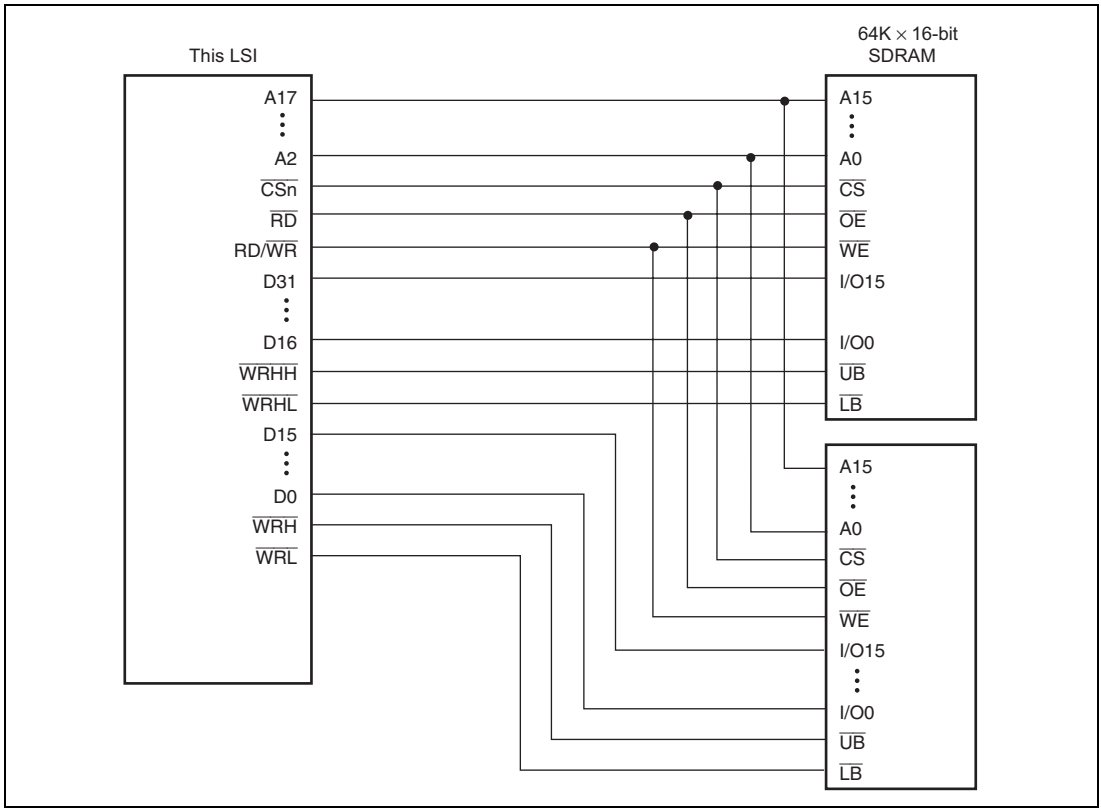


Figure 9.40 Example of Connection with 32-Bit Data-Width SRAM with Byte Selection

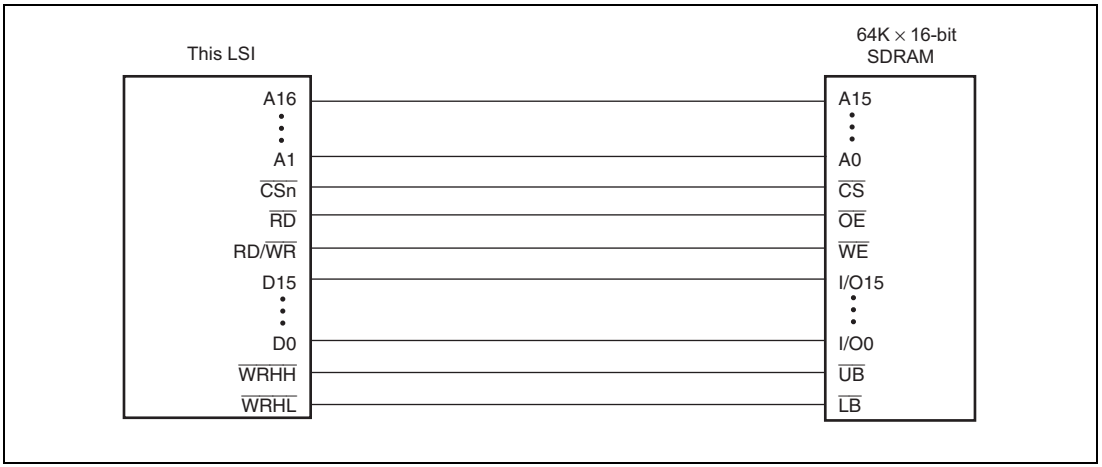


Figure 9.41 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

9.5.9 Burst ROM (Clock Synchronous) Interface

The burst ROM (clock synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM (clock synchronous) is accessed, the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, using 16-byte read by the DMA is recommended. The burst ROM interface performs write access in the same way as normal space access.

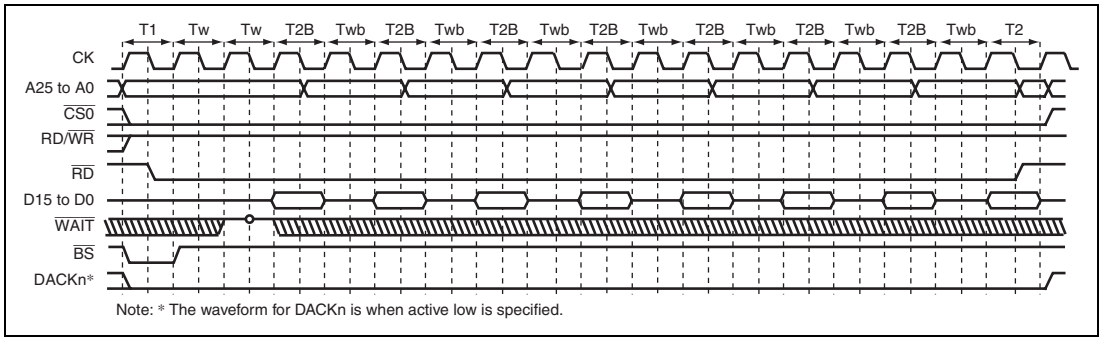


Figure 9.42 Burst ROM Access Timing (Clock Synchronous)
(Burst Length = 8, Wait Cycles Inserted in First Access = 2,
Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

9.5.10 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. A function that avoids data collisions by inserting idle (wait) cycles between continuous access cycles has been newly added.

The number of wait cycles between access cycles can be set by the WM bit in CSnWCR, bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR, and bits DMAIW2 to DMAIW0 and DMAIWA in CMNCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space
6. Data output from an external device caused by DMA single address transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single address transfer is followed by any type of access (DMAIWA = 1)

For the specification of the number of idle cycles between access cycles described above, refer to the description of each register.

Besides the idle cycles between access cycles specified by the registers, idle cycles must be inserted to interface with the internal bus or to obtain the minimum pulse width for a multiplexed pin ($\overline{\text{WRxx}}$). The following gives detailed information about the idle cycles and describes how to estimate the number of idle cycles.

The number of idle cycles on the external bus from $\overline{\text{CSn}}$ negation to $\overline{\text{CSn}}$ or $\overline{\text{CSm}}$ assertion is described below.

There are eight conditions that determine the number of idle cycles on the external bus as shown in table 9.21. The effects of these conditions are shown in figure 9.43.

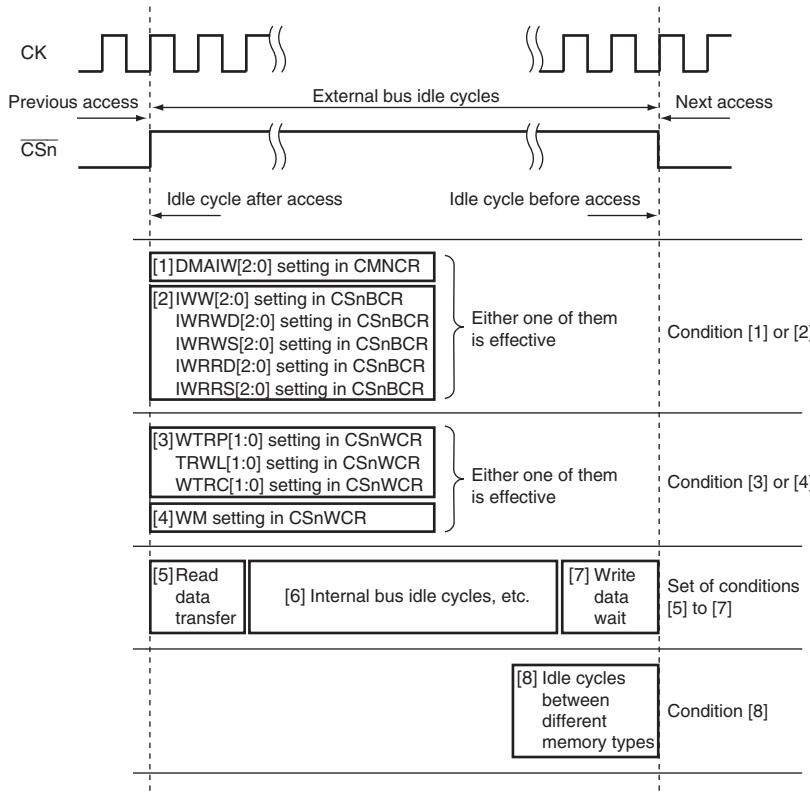
Table 9.21 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
(1)	DMAIW[2:0] in CMNCR	These bits specify the number of idle cycles for DMA single address transfer. This condition is effective only for single address transfer and generates idle cycles after the access is completed.	0 to 12	When 0 is specified for the number of idle cycles, the DACK signal may be asserted continuously. This causes a discrepancy between the number of cycles detected by the device with DACK and the DMAC transfer count, resulting in a malfunction.
(2)	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access other than single address transfer. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition is effective only for access cycles other than single address transfer and generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
(3)	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait cycles and idle cycles between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle cycles after the access is completed	0 to 3	Specify these bits in accordance with the specification of the target SDRAM.
(4)	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	

No.	Condition	Description	Range	Note
(5)	Read data transfer cycle	One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not B'00.	0 or 1*	One idle cycle is always generated after a read cycle with SDRAM interface.
(6)	Internal bus idle cycles, etc.	External bus access requests from the CPU or DMAC and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the BSC when the access size is larger than the external data bus width.	0 or larger	The number of internal bus idle cycles may not become 0 depending on the Iϕ:Bϕ clock ratio. Tables 9.22 and 9.23 show the relationship between the clock ratio and the minimum number of internal bus idle cycles.
(7)	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles are frequently available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through CSnBCR.
(8)	Idle cycles between different memory types	To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2.5	The number of idle cycles depends on the target memory types. See table 9.24.

Note: * This is the case for consecutive read operations when the data read are stored in separate registers.

In the above conditions, a total of four conditions, that is, condition (1) or (2) (either one is effective), condition (3) or (4) (either one is effective), a set of conditions (5) to (7) (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition (8) are generated at the same time. The maximum number of idle cycles among these four conditions becomes the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition (1) or (2).



Note: A total of four conditions (condition [1] or [2], condition [3] or [4], a set of conditions [5] to [7], and condition [8]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 9.43 Idle Cycle Conditions

Table 9.22 Minimum Number of Idle Cycles on Internal Bus (CPU Operation)

	Clock Ratio (Iϕ:Bϕ)			
CPU Operation	**8:1**	**4:1**	**2:1**	**1:1**
Write → write	0	0	0	0
Write → read	0	0	0	0
Read → write	1	1	2	3
Read → read	0	0	0	0

Conditions:

- The bits for setting the idle cycles between access cycles in CS1BCR and CS2BCR are all set to 0.
- In CS1WCR and CS2WCR, the WM bit is set to 1 (external $\overline{\text{WAIT}}$ pin disabled) and the HW[1:0] bits are set to 00 ($\overline{\text{CS}}$ negation is not extended).
- For both the CS1 and CS2 spaces, normal SRAM devices are connected, the bit width is 32 bits, and access size is also 32 bits.

Table 9.23 Minimum Number of Idle Cycles on Internal Bus (DMAC Operation)

	Transfer Mode					
DMAC Operation	**Dual Address**				**Single Address*2**	
Activation source	**Auto request**	**Peripheral module request**	**External request (level)**	**External request (edge)**	**External request (level)**	**External request (edge)**
Write → write	1	1	3	3	6	1
Write → read	0	0	2 or 0*1	1 or 0*1	0	0
Read → write	0	0	0	0	0	0
Read → read	2	2	5	4	5	2

Operating conditions:

1. The write → write cycle means transfer from an on-chip memory to an external memory. The read → read cycle means transfer from an external memory to an on-chip memory. The write → read cycle and read → write cycle mean transfer between external memories. Each of the operations is performed in burst mode.
2. The external data bus width is 16 bits and the DMA transfer size is 16 bits.
3. Ick : Bck = 1 : 1/4

Notes: 1. For the write → read cycles in transfer with an external request (level), 0 means different channels are activated successively and 2 means the same channel is activated successively.
For the write → read cycles in transfer with an external request (edge), 0 means different channels are activated successively and 1 means the same channel is activated successively.
2. The write → read and read → write columns in single address transfer indicate the case when different channels are activated successively. The "write" means transfer from a device with DACK to external memory and the "read" means transfer from external memory to a device with DACK.

Table 9.24 Number of Idle Cycles Inserted between Access Cycles to Different Memory Types

	Next Cycle							
Previous Cycle	**SRAM**	**Burst ROM (Asynchronous)**	**MPX-I/O**	**Byte SRAM (BAS = 0)**	**Byte SRAM (BAS = 1)**	**SDRAM**	**SDRAM (Low-Frequency Mode)**	**Burst ROM (Synchronous)**
SRAM	0	0	1	0	1	1	1.5	0
Burst ROM (asynchronous)	0	0	1	0	1	1	1.5	0
MPX-I/O	1	1	0	1	1	1	1.5	1
Byte SRAM (BAS = 0)	0	0	1	0	1	1	1.5	0
Byte SRAM (BAS = 1)	1	1	2	1	0	0	1.5	1
SDRAM	1	1	2	1	0	0	—	1
SDRAM (low-frequency mode)	1.5	1.5	2.5	1.5	0.5	—	1	1.5
Burst ROM (synchronous)	0	0	1	0	1	1	1.5	0

Figure 9.43 shows sample estimation of idle cycles between access cycles. In the actual operation, the idle cycles may become shorter than the estimated value due to the write buffer effect or may become longer due to internal bus idle cycles caused by stalling in the pipeline due to CPU instruction execution or CPU register conflicts. Please consider these errors when estimating the idle cycles.

Sample Estimation of Idle Cycles between Access Cycles

This example estimates the idle cycles for data transfer from the CS1 space to CS2 space by CPU access. Transfer is repeated in the following order: CS1 read → CS1 read → CS2 write → CS2 write → CS1 read → ...

- Conditions

The bits for setting the idle cycles between access cycles in CS1BCR and CS2BCR are all set to 0.
In CS1WCR and CS2WCR, the WM bit is set to 1 (external $\overline{\text{WAIT}}$ pin disabled) and the HW[1:0] bits are set to 00 ($\overline{\text{CS}}$ negation is not extended).
Iϕ:Bϕ is set to 4:1, and no other processing is done during transfer.
For both the CS1 and CS2 spaces, normal SRAM devices are connected, the bus width is 32 bits, and access size is also 32 bits.

The idle cycles generated under each condition are estimated for each pair of access cycles. In the following table, R indicates a read cycle and W indicates a write cycle.

Condition	R → R	R → W	W → W	W → R	Note
[1] or [2]	0	0	0	0	CSnBCR is set to 0.
[3] or [4]	0	0	0	0	The WM bit is set to 1.
[5]	1	1	0	0	Generated after a read cycle.
[6]	0	1	0	0	See the Iϕ:Bϕ = 4:1 column in table 9.22.
[7]	0	1	0	0	No idle cycle is generated for the second time due to the write buffer effect.
[5] + [6] + [7]	1	3	0	0	
[8]	0	0	0	0	Value for SRAM → SRAM access
Estimated idle cycles	1	3	0	0	Maximum value among conditions [1] or [2], [3] or [4], [5] + [6] + [7], and [8]
Actual idle cycles	1	3	0	1	The estimated value does not match the actual value in the W → R cycles because the internal idle cycles due to condition [6] is estimated as 0 but actually an internal idle cycle is generated due to execution of a loop condition check instruction.

Figure 9.44 Comparison between Estimated Idle Cycles and Actual Value

9.5.11 Bus Arbitration

In bus arbitration by this LSI, it normally holds bus mastership but can release this after receiving a bus request from another device.

Bus arbitration by this LSI also supports four on-chip bus masters: the CPU, DMAC, DTC, and EDMAC. The priority order of these bus masters is as follows.

Bus mastership request from an external device ($\overline{\text{BREQ}}$) > EDMAC > DTC > DMAC > CPU.

Bus mastership is transferred at the boundary of bus cycles. Namely, bus mastership is released immediately after receiving a bus request when a bus cycle is not being performed. The release of bus mastership is delayed until the bus cycle is complete when a bus cycle is in progress. Even when from outside the LSI it looks like a bus cycle is not being performed, a bus cycle may be performing internally, started by inserting wait cycles between access cycles. Therefore, it cannot be immediately determined whether or not bus mastership has been released by looking at the $\overline{\text{CS}}$n signal or other bus control signals. The states that do not allow bus mastership release are shown below.

1. Between the read and write cycles of a TAS instruction, or 64-bit transfer cycle of an FMOV instruction
2. Multiple bus cycles generated when the data bus width is smaller than the access size (for example, between bus cycles when longword access is made to a memory with a data bus width of 8 bits)
3. 16-byte transfer by the DMAC
4. Setting the BLOCK bit in CMNCR to 1

Moreover, by using DPRTY bit in CMNCR, whether the bus mastership request is received or not can be selected during DMAC burst transfer.

The LSI has the bus mastership until a bus request is received from another device. Upon acknowledging the assertion (low level) of the external bus request signal $\overline{\text{BREQ}}$, the LSI releases the bus at the completion of the current bus cycle and asserts the $\overline{\text{BACK}}$ signal. After the LSI acknowledges the negation (high level) of the $\overline{\text{BREQ}}$ signal that indicates the external device has released the bus, it negates the $\overline{\text{BACK}}$ signal and resumes the bus usage.

With the SDRAM interface, all bank pre-charge commands (PALLs) are issued when active banks exist and the bus is released after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CK. The bus mastership enable signal is asserted 0.5 cycles

after the above timing, synchronized with the falling edge of CK. The bus control signals ($\overline{BS}$, $\overline{CSn}$, $\overline{RASL}$, $\overline{CASL}$, CKE, DQMxx, $\overline{WRxx}$, $\overline{RD}$, and RD/$\overline{WR}$) are placed in the high-impedance state at subsequent rising edges of CK. Bus request signals are sampled at the falling edge of CKIO. Note that CKE, $\overline{RASL}$, and $\overline{CASL}$ can continue to be driven at the previous value even in the bus-released state by setting the HIZCNT bit in CMNCR.

The sequence for reclaiming the bus mastership from an external device is described below. 1.5 cycles after the negation of $\overline{BREQ}$ is detected at the falling edge of CK, the bus control signals are driven high. The bus acknowledge signal is negated at the next falling edge of the clock. The fastest timing at which actual bus cycles can be resumed after bus control signal assertion is at the rising edge of the CK where address and data signals are driven. Figure 9.44 shows the bus arbitration timing.

When it is necessary to refresh SDRAM while releasing the bus mastership, the bus mastership should be returned using the $\overline{REFOUT}$ signal. For details on the selection of $\overline{REFOUT}$, see section 22, Pin Function Controller (PFC). The $\overline{REFOUT}$ signal is kept asserting at low level until the bus mastership is acquired. The $\overline{BREQ}$ signal is negated by asserting the $\overline{REFOUT}$ signal and the bus mastership is returned from the external device. If the bus mastership is not returned for a refreshing period or longer, the contents of SDRAM cannot be guaranteed because a refreshing cannot be executed.

While releasing the bus mastership, the SLEEP instruction (to enter sleep mode or standby mode), as well as a manual reset, cannot be executed until the LSI obtains the bus mastership.

The $\overline{BREQ}$ input signal is ignored in standby mode and the $\overline{BACK}$ output signal is placed in the high impedance state. If the bus mastership request is required in this state, the bus mastership must be released by pulling down the $\overline{BACK}$ pin to enter standby mode.

The bus mastership release ($\overline{BREQ}$ signal for high level negation) after the bus mastership request ($\overline{BREQ}$ signal for low level assertion) must be performed after the bus usage permission ($\overline{BACK}$ signal for low level assertion). If the $\overline{BREQ}$ signal is negated before the $\overline{BACK}$ signal is asserted, only one cycle of the $\overline{BACK}$ signal is asserted depending on the timing of the $\overline{BREQ}$ signal to be negated and this may cause a bus contention between the external device and the LSI.

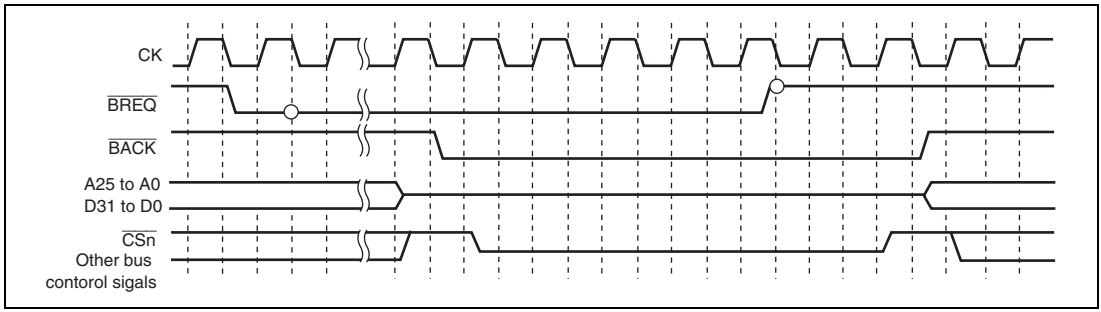


Figure 9.45 Bus Arbitration Timing

9.5.12 Others

(1) Reset

The bus state controller (BSC) can be initialized completely only at power-on reset. At power-on reset, all signals are negated and data output buffers are turned off regardless of the bus cycle state after the internal reset is synchronized with the internal clock. All control registers are initialized. In standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, only the current bus cycle being executed is completed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle.

(2) Access from the Side of the LSI Internal Bus Master

Since the bus state controller (BSC) incorporates a four-stage write buffer, the BSC can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the BSC functions in the same way for an access by a bus master other than the CPU such as the DMAC. Accordingly, to perform dual address DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next write cycle will not be initiated until the previous write cycle is completed.

Changing the registers in the BSC while the write buffer is operating may disrupt correct write access. Therefore, do not change the registers in the BSC immediately after a write access. If this change becomes necessary, do it after executing a dummy read of the write data.

(3) On-Chip Peripheral Module Access

To access an on-chip module register, two or more peripheral module clock (Pϕ) cycles are required. Care must be taken in system design.

When the CPU writes data to the internal peripheral registers, the CPU performs the succeeding instructions without waiting for the completion of writing to registers.

For example, a case is described here in which the system is transferring to software standby mode for power savings. To make this transition, the SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR register is indispensable to complete writing to the STBY bit.

To reflect the change by internal peripheral registers while performing the succeeding instructions, execute a dummy read of registers to which write instruction is given and then perform the succeeding instructions.

Table 9.25 shows the number of cycles required for access to the on-chip peripheral I/O registers by the CPU.

Table 9.25 Number of Cycles for Access to On-Chip Peripheral Module Registers

	Number of Access Cycles	Remarks
Write	$(2 + n) \times I\phi + (1 + m) \times B\phi + 2 \times P\phi$	Except for the FLD, E-DMAC, and EtherC
	$(2 + n) \times I\phi + (1 + m) \times B\phi + 3 \times P\phi$	FLD access
	$(2 + n) \times I\phi + 3 \times B\phi$	E-DMAC access
	$(2 + n) \times I\phi + 9 \times B\phi$	EtherC access
Read	$(2 + n) \times I\phi + (1 + m) \times B\phi + 2 \times P\phi + (2 + l) \times I\phi$	Except for the FLD, E-DMAC, and EtherC
	$(2 + n) \times I\phi + (1 + m) \times B\phi + 3 \times P\phi + (2 + l) \times I\phi$	FLD access
	$(2 + n) \times I\phi + 4 \times B\phi + (2 + l) \times I\phi$	E-DMAC access
	$(2 + n) \times I\phi + 12 \times B\phi + (2 + l) \times I\phi$	EtherC access

Notes: The above indicates the number of access cycles of which executed when the instructions are by on-chip ROM or by on-chip RAM.
When Iϕ:Bϕ = 1:1, n = 0 and l = 0.
When Iϕ:Bϕ = 2:1, n = 1 to 0 and l = 0.
When Iϕ:Bϕ = 4:1, n = 3 to 0 and l = 0, 1.

When Iϕ:Bϕ = 8:1, n = 7 to 0 and I = 1.
When Bϕ:Pϕ = 1:1, m = 0.
When Bϕ:Pϕ = 2:1, m = 1, 0.
n and m depend on the internal execution state.

Synchronous logic and a layered bus structure have been adopted for this LSI. Data on each bus are input and output in synchronization with rising edges of the corresponding clock signal. The C bus, the I bus, and the peripheral bus are synchronized with the Iϕ, Bϕ, and Pϕ clock, respectively.

Figure 9.45 shows an example of the timing of write access to the peripheral bus when Iϕ:Bϕ:Pϕ = 4:1:1. Data are output to the C bus, which is connected to the CPU, in synchronization with Iϕ. When Iϕ:Bϕ = 4:1, there are 4 cycles of this clock to one cycle of Bϕ, so four transfers to the I bus can proceed in one cycle of Bϕ. Thus, a period of up to 5 × Iϕ may be required before a rising edge of Bϕ, which is the time of transfer from the C bus to the I bus (a case where this takes 3 cycles of Iϕ is indicated in figure 9.45). When Iϕ: Bϕ = 4:1, transfer of data from the C bus to the I bus takes (2 + n) × Iϕ (n = 0 to 3). The relation between the timing of data transfer to the C bus and the rising edge of Bϕ depends on the state of program execution. When Bϕ:Pϕ = 1:1, transfer of data from the I bus to the peripheral bus takes 1Bϕ + 2Pϕ. In the case shown in the figure, where n = 1 and m = 0, the time required for access is 3 × Iϕ + 2 × Bϕ + 2 × Pϕ.

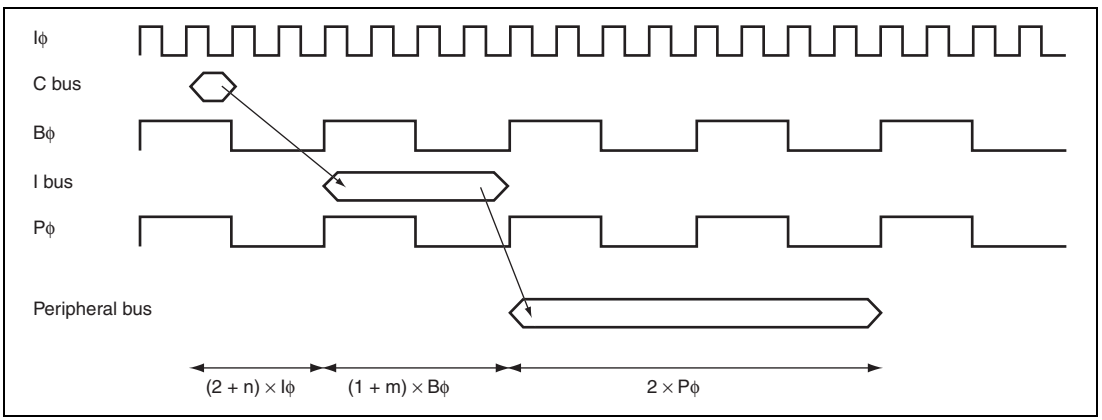


Figure 9.46 Timing of Write Access to On-Chip Peripheral I/O Registers When Iϕ;Bϕ:Pϕ = 4:1:1

Figure 9.46 shows an example of timing of read access to the peripheral bus when Iϕ:Bϕ:Pϕ = 4:2:1. Transfer from the C bus to the peripheral bus is performed in the same way as for write access. In the case of reading, however, values output onto the peripheral bus must be transferred to the CPU. Although transfers from the peripheral bus to the I bus and from the I bus to the C bus are performed in synchronization with the rising edge of the respective bus clocks, a period of (2 + l) × Iϕ is actually required because Iϕ ≥ Bϕ≥ Pϕ. In the case shown in the figure 9.46, where n = 1, m = 1, and l = 1, the time required for access is 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 3 × Iϕ.

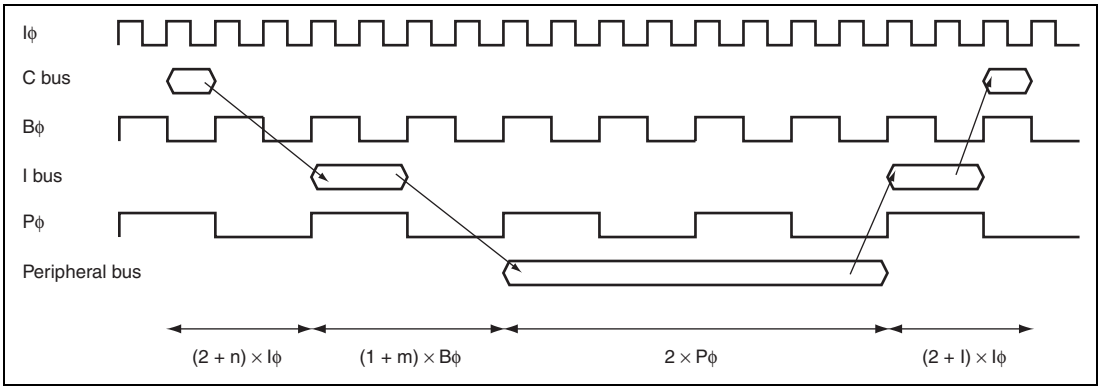


Figure 9.47 Timing of Read Access to On-Chip Peripheral I/O Registers When Iϕ:Bϕ:Pϕ = 4:2:1

Note that the peripheral bus cycle for the FLD is different from that for other modules. The cycle for the FLD is 3 × Pϕ.

(4) Access to On-Chip Memory and External Device

Table 9.26 shows the number of cycles required for access to the on-chip memory and external device.

Table 9.26 Number of Cycles for Access to On-Chip Memory and External Device

Object to be accessed		128-bit on-chip ROM (2Iϕ)	On-chip RAM (Iϕ)		External device (Bϕ)*5		
			Pages 0 to 3	Pages 4 to 7			
		ROM cache (Iϕ)	(high speed)	(low speed)			
Bus width		**—**	**32 bits**	**32 bits**	**8 bits**	**16 bits**	**32 bits**
Access from CPU	Instruction fetch	1 to 3	1	2	(2 + n) × Iϕ + (3 + m) × Bϕ + 3×(2 + o) × Bϕ + (2 + I) × Iϕ	(2 + n) × Iϕ + (3 + m) × Bϕ + 1 × (2 + o) × Bϕ + (2 + I) × Iϕ	
	Data read (longword)	1 to 3	1	2			
	Data read (word)	1 to 3	1	2	(2 + n) × Iϕ + (3 + m) × Bϕ + 1 × (2 + o) × Bϕ + (2 + I) × Iϕ	(2 + n) × Iϕ + (3 + m) × Bϕ + (2 + I) × Iϕ	(2 + n) × Iϕ + (3 + m) × Bϕ + (2 + I) × Iϕ
	Data read (byte)	1 to 3	1	2	(2 + n) × Iϕ + (3 + m) × Bϕ + (2 + I) × Iϕ		
	Data write*1 (longword)	—	1	3	(2 + n) × Iϕ + (4 + m) × Bϕ + 3 × (2 + o) × Bϕ	(2 + n) × Iϕ + (4 + m) × Bϕ + 1 × (2 + o) × Bϕ	
	Data write*1 (word)	—	1	3	(2 + n) × Iϕ + (4 + m) × Bϕ + 1 × (2 + o) × Bϕ	(2 + n) × Iϕ + (4 + m) × Bϕ	(2 + n) × Iϕ + (4 + m) × Bϕ
	Data read (byte)	—	1	3	(2 + n) × Iϕ + (4 + m) × Bϕ +		
Access from modules other than CPU	Data read (longword)	3Bϕ to 4Iϕ + 3Bϕ*2	1Bϕ to 4Bϕ*3		9Bϕ	5Bϕ	3Bϕ
	Data read (word)				5Bϕ	3Bϕ	
	Data read (byte)				3Bϕ		
	Data write*1 (longword)	—	1Bϕ to 3Bϕ*4		9Bϕ	5Bϕ	3Bϕ
	Data write*1 (word)	—			5Bϕ	3Bϕ	
	Data read (byte)	—			3Bϕ		

Notes:
1. Using the write buffer, the bus master can execute the succeeding processing before he previous write cycle is completed. For details, see section 9.5.12 (2), Access from he Side of the LSI Internal Bus Master.
2. When Iϕ:Bϕ = 1:1/8, the value is 3Bϕ. When Iϕ:Bϕ is not 1:1/8, the value is 4Iϕ + 3Bϕ.
3. When Iϕ:Bϕ = 8:1, the value is 1Bϕ. When Iϕ:Bϕ = 4:1, the value is 2Bϕ. When Iϕ:Bϕ = 2:1, the value is 2Bϕ to 3Bϕ. When Iϕ:Bϕ = 1:1, the value is 3Bϕ to 4Bϕ.
4. When Iϕ:Bϕ = 8:1, the value is 1Bϕ. When Iϕ:Bϕ = 4:1, the value is 1Bϕ to 2Bϕ. When Iϕ:Bϕ = 2:1, the value is 2Bϕ. When Iϕ:Bϕ = 1:1, the value is 2Bϕ to 3Bϕ.
5. The above indicates the number of access cycles of which executed when the instructions are by on-chip ROM or by on-chip RAM.

   When Iϕ:Bϕ = 1:1, n = 0 and I = 0.

   When Iϕ:Bϕ = 2:1, n = 1 to 0 and I = 0.

   When Iϕ:Bϕ = 4:1, n = 3 to 0 and I = 0, 1.

   When Iϕ:Bϕ = 8:1, n = 7 to 0 and I = 1.

   m = wait cycle

   o = idle cycle + wait cycle

   n and I depend on the internal execution state.

Figure 9.48shows an example of timing of write access to the longword data and word data which are twice as much as 16- and 8-bit external bus widths, respectively, in the external memory when Iϕ:Bϕ = 2:1.

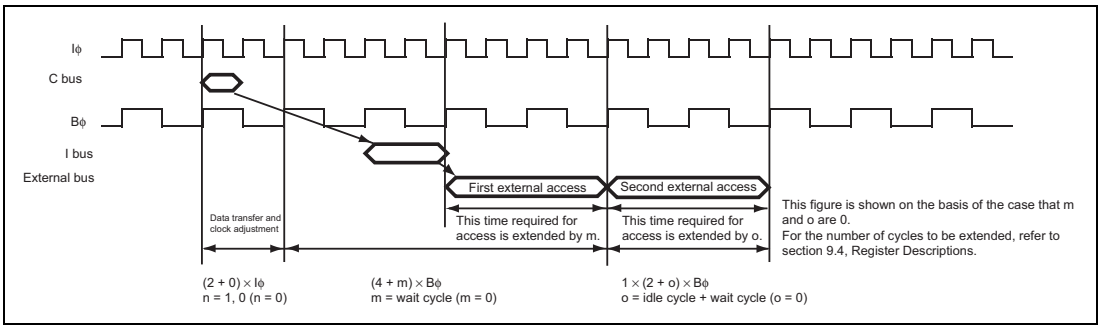


Figure 9.48 Timing of Write Access to Data Beyond External Bus Width When Iϕ:Bϕ = 2:1

Figure 9.49 shows an example of timing of read access to the data within the external bus width from the external memory when Iϕ:Bϕ = 4:1.

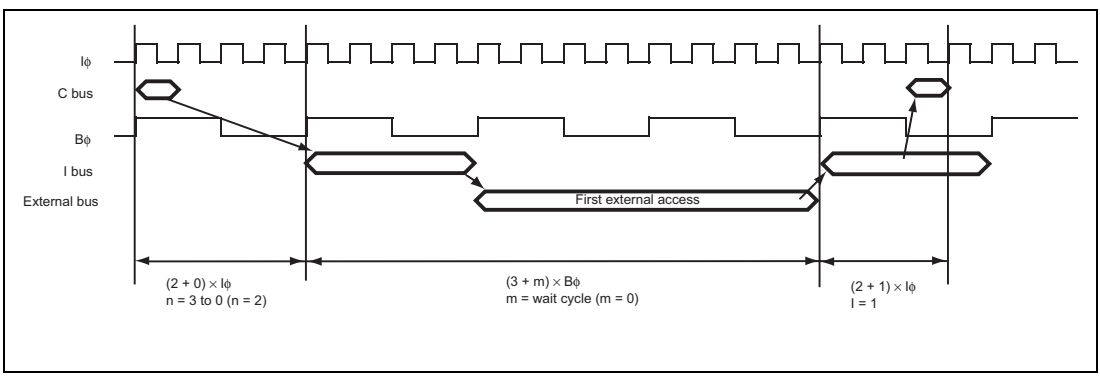


Figure 9.49 Timing of Read Access to Data within External Bus Width When Iϕ:Bϕ = 4:1

9.6 Interrupt Source

The BSC has the compare match interrupt (CMI) as an interrupt source.

Table 9.26 gives details on this interrupt source. The compare match interrupt enable bit (CMIE) in the refresh timer control/status register (RTCSR) can be used to enable or disable the interrupt source.

The compare match interrupt (CMI) is generated when the compare match flag (CMF) and compare match interrupt enable bit (CMIE) in RTCSR are set to 1.

Clearing the interrupt flag bit to 0 cancels the interrupt request.

Table 9.26 Interrupt Source

Abbreviation	Interrupt Source	Interrupt Enable Bit	Interrupt Flag
CMI	Compare match interrupt	CMIE	CMIF

9.7 Usage Note

9.7.1 Note on Connection of External LSI Circuits such as SDRAMs and ASICs

Each of the following pairs of pin functions among the pins for the output of SDRAM control signals is multiplexed on respective single pins: RD/$\overline{WR}$ and A23, $\overline{RASL}$ and A18, and $\overline{CASL}$ and A19. When an external chip (SDRAM, ASIC, etc.) is to be connected to the bus of this LSI, follow the procedure described below.

- Use the 23 bits from A0 to A22 as the address for the external chip such as ASICs.

Section 10 Direct Memory Access Controller (DMAC)

The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

10.1 Features

- Number of channels selectable: Eight channels (channels 0 to 7) max.
  CH0 to CH3 channels can only receive external requests.
- 4-Gbyte physical address space
- Transfer data length is selectable: Byte, word (two bytes), longword (four bytes), and 16 bytes (longword × 4)
- Maximum transfer count: 16,777,216 transfers (24 bits)
- Address mode: Dual address mode and single address mode are supported.
- Transfer requests
  — External request
  — On-chip peripheral module request
  — Auto request
  The following modules can issue on-chip peripheral module requests.
  — Two SCIF sources, two IIC3 sources, one A/D converter source, five MTU2 sources, two CMT sources, four USB sources, two RSPI sources, and one RCAN-ET source
- Selectable bus modes
  — Cycle steal mode (normal mode and intermittent mode)
  — Burst mode
- Selectable channel priority levels: The channel priority levels are selectable between fixed mode and round-robin mode.
- Interrupt request: An interrupt request can be sent to the CPU on completion of half- or full-data transfer. Through the HE and HIE bits in CHCR, an interrupt is specified to be issued to the CPU when half of the initially specified DMA transfer is completed.
- External request detection: There are following four types of DREQ input detection.
  — Low level detection
  — High level detection
  — Rising edge detection
  — Falling edge detection

- Transfer request acknowledge and transfer end signals: Active levels for DACK and TEND can be set independently.
- Support of reload functions in DMA transfer information registers: DMA transfer using the same information as the current transfer can be repeated automatically without specifying the information again. Modifying the reload registers during DMA transfer enables next DMA transfer to be done using different transfer information. The reload function can be enabled or disabled independently in each channel.

Figure 10.1 shows the block diagram of the DMAC.

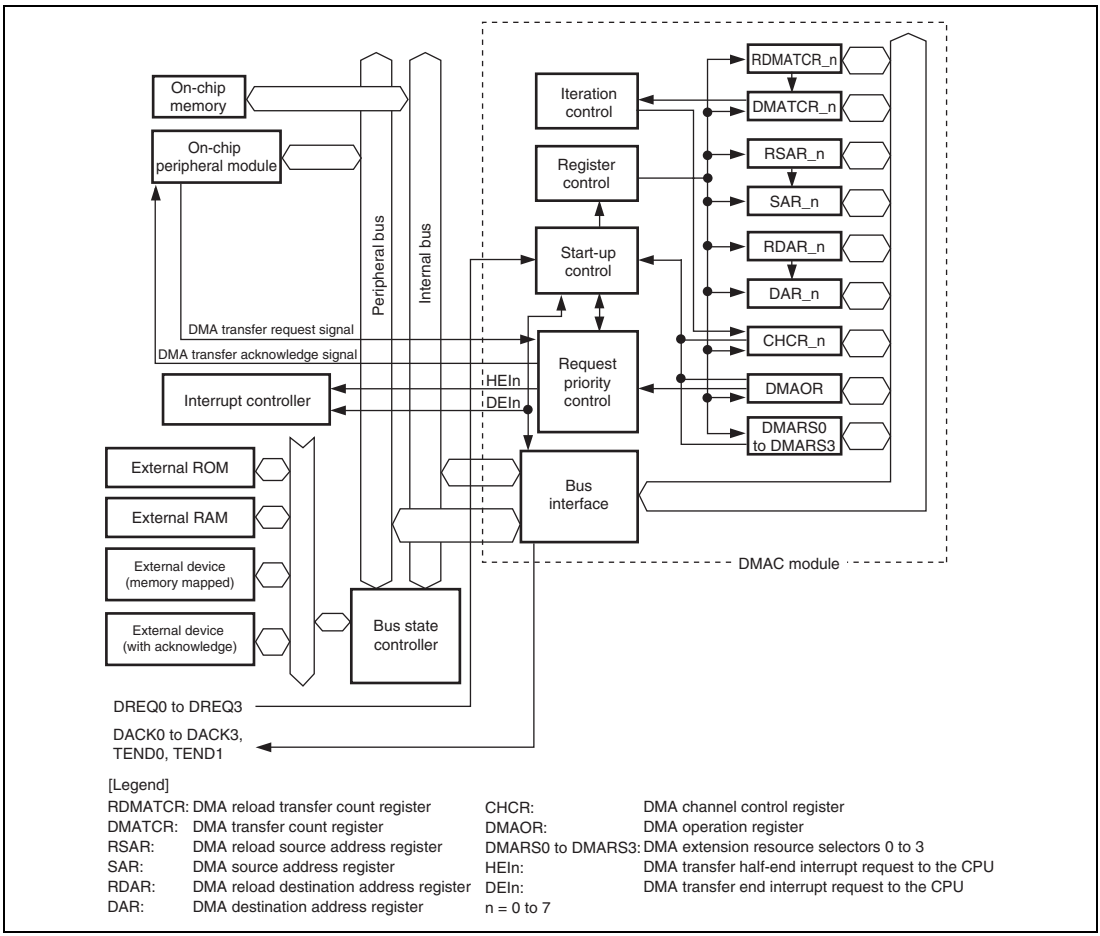


Figure 10.1 Block Diagram of DMAC

10.2 Input/Output Pins

The external pins for DMAC are described below. Table 10.1 lists the configuration of the pins that are connected to external bus. DMAC has pins for four channels (CH0 to CH3) as the external bus use.

Table 10.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	DMA transfer request	DREQ0	I	DMA transfer request input from an external device to channel 0
	DMA transfer request acknowledge	DACK0	O	DMA transfer request acknowledge output from channel 0 to an external device
1	DMA transfer request	DREQ1	I	DMA transfer request input from an external device to channel 1
	DMA transfer request acknowledge	DACK1	O	DMA transfer request acknowledge output from channel 1 to an external device
2	DMA transfer request	DREQ2	I	DMA transfer request input from an external device to channel 2
	DMA transfer request acknowledge	DACK2	O	DMA transfer request acknowledge output from channel 2 to an external device
3	DMA transfer request	DREQ3	I	DMA transfer request input from an external device to channel 3
	DMA transfer request acknowledge	DACK3	O	DMA transfer request acknowledge output from channel 3 to an external device
0	DMA transfer end	TEND0	O	DMA transfer end output for channel 0
1	DMA transfer end	TEND1	O	DMA transfer end output for channel 1

10.3 Register Descriptions

The DMAC has the registers listed in table 10.2. There are four control registers and three reload registers for each channel, and one common control register is used by all channels. In addition, there is one extension resource selector per two channels. Each channel number is expressed in the register names, as in SAR_0 for SAR in channel 0.

Table 10.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA source address register_0	SAR_0	R/W	H'00000000	H'FFFE1000	16, 32
	DMA destination address register_0	DAR_0	R/W	H'00000000	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR_0	R/W	H'00000000	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR_0	R/W*1	H'00000000	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR_0	R/W	H'00000000	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR_0	R/W	H'00000000	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR_0	R/W	H'00000000	H'FFFE1108	16, 32
1	DMA source address register_1	SAR_1	R/W	H'00000000	H'FFFE1010	16, 32
	DMA destination address register_1	DAR_1	R/W	H'00000000	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR_1	R/W	H'00000000	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR_1	R/W*1	H'00000000	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR_1	R/W	H'00000000	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR_1	R/W	H'00000000	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR_1	R/W	H'00000000	H'FFFE1118	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	DMA source address register_2	SAR_2	R/W	H'00000000	H'FFFE1020	16, 32
	DMA destination address register_2	DAR_2	R/W	H'00000000	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR_2	R/W	H'00000000	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR_2	R/W*1	H'00000000	H'FFFE102C	8, 16, 32
	DMA reload source address register_2	RSAR_2	R/W	H'00000000	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR_2	R/W	H'00000000	H'FFFE1124	16, 32
	DMA reload transfer count register_2	RDMATCR_2	R/W	H'00000000	H'FFFE1128	16, 32
3	DMA source address register_3	SAR_3	R/W	H'00000000	H'FFFE1030	16, 32
	DMA destination address register_3	DAR_3	R/W	H'00000000	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR_3	R/W	H'00000000	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR_3	R/W*1	H'00000000	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR_3	R/W	H'00000000	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR_3	R/W	H'00000000	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR_3	R/W	H'00000000	H'FFFE1138	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	DMA source address register_4	SAR_4	R/W	H'00000000	H'FFFE1040	16, 32
	DMA destination address register_4	DAR_4	R/W	H'00000000	H'FFFE1044	16, 32
	DMA transfer count register_4	DMATCR_4	R/W	H'00000000	H'FFFE1048	16, 32
	DMA channel control register_4	CHCR_4	R/W*1	H'00000000	H'FFFE104C	8, 16, 32
	DMA reload source address register_4	RSAR_4	R/W	H'00000000	H'FFFE1140	16, 32
	DMA reload destination address register_4	RDAR_4	R/W	H'00000000	H'FFFE1144	16, 32
	DMA reload transfer count register_4	RDMATCR_4	R/W	H'00000000	H'FFFE1148	16, 32
5	DMA source address register_5	SAR_5	R/W	H'00000000	H'FFFE1050	16, 32
	DMA destination address register_5	DAR_5	R/W	H'00000000	H'FFFE1054	16, 32
	DMA transfer count register_5	DMATCR_5	R/W	H'00000000	H'FFFE1058	16, 32
	DMA channel control register_5	CHCR_5	R/W*1	H'00000000	H'FFFE105C	8, 16, 32
	DMA reload source address register_5	RSAR_5	R/W	H'00000000	H'FFFE1150	16, 32
	DMA reload destination address register_5	RDAR_5	R/W	H'00000000	H'FFFE1154	16, 32
	DMA reload transfer count register_5	RDMATCR_5	R/W	H'00000000	H'FFFE1158	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	DMA source address register_6	SAR_6	R/W	H'00000000	H'FFFE1060	16, 32
	DMA destination address register_6	DAR_6	R/W	H'00000000	H'FFFE1064	16, 32
	DMA transfer count register_6	DMATCR_6	R/W	H'00000000	H'FFFE1068	16, 32
	DMA channel control register_6	CHCR_6	R/W*1	H'00000000	H'FFFE106C	8, 16, 32
	DMA reload source address register_6	RSAR_6	R/W	H'00000000	H'FFFE1160	16, 32
	DMA reload destination address register_6	RDAR_6	R/W	H'00000000	H'FFFE1164	16, 32
	DMA reload transfer count register_6	RDMATCR_6	R/W	H'00000000	H'FFFE1168	16, 32
7	DMA source address register_7	SAR_7	R/W	H'00000000	H'FFFE1070	16, 32
	DMA destination address register_7	DAR_7	R/W	H'00000000	H'FFFE1074	16, 32
	DMA transfer count register_7	DMATCR_7	R/W	H'00000000	H'FFFE1078	16, 32
	DMA channel control register_7	CHCR_7	R/W*1	H'00000000	H'FFFE107C	8, 16, 32
	DMA reload source address register_7	RSAR_7	R/W	H'00000000	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR_7	R/W	H'00000000	H'FFFE1174	16, 32
	DMA reload transfer count register_7	RDMATCR_7	R/W	H'00000000	H'FFFE1178	16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	DMA operation register	DMAOR	R/W*2	H'0000	H'FFFE1200	8, 16
0 and 1	DMA extension resource selector 0	DMARS0	R/W	H'0000	H'FFFE1300	16
2 and 3	DMA extension resource selector 1	DMARS1	R/W	H'0000	H'FFFE1304	16
4 and 5	DMA extension resource selector 2	DMARS2	R/W	H'0000	H'FFFE1308	16
6 and 7	DMA extension resource selector 3	DMARS3	R/W	H'0000	H'FFFE130C	16

Notes: 1. For the HE and TE bits in CHCRn, only 0 can be written to clear the flags after 1 is read.
2. For the AE and NMIF bits in DMAOR, only 0 can be written to clear the flags after 1 is read.

10.3.1 DMA Source Address Registers (SAR)

The DMA source address registers (SAR) are 32-bit readable/writable registers that specify the source address of a DMA transfer. During a DMA transfer, these registers indicate the next source address. When the data of an external device with DACK is transferred in single address mode, SAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

SAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.2 DMA Destination Address Registers (DAR)

The DMA destination address registers (DAR) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. During a DMA transfer, these registers indicate the next destination address. When the data of an external device with DACK is transferred in single address mode, DAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

DAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.3 DMA Transfer Count Registers (DMATCR)

The DMA transfer count registers (DMATCR) are 32-bit readable/writable registers that specify the number of DMA transfers. The transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. During a DMA transfer, these registers indicate the remaining transfer count.

The upper eight bits of DMATCR are always read as 0, and the write value should always be 0. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

DMATCR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.4 DMA Channel Control Registers (CHCR)

The DMA channel control registers (CHCR) are 32-bit readable/writable registers that control DMA transfer mode.

The DO, AM, AL, DL, and DS bits which specify the DREQ and DACK external pin functions can be read and written to in channels 0 to 3, but they are reserved in channels 4 to 7. The TL bit which specifies the TEND external pin function can be read and written to in channels 0 and 1, but it is reserved in channels 2 to 7. Before modifying the CHCR setting, clear the DE bit for the corresponding channel.

CHCR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC	-	-	RLD	-	-	-	-	DO	TL	-	-	HE	HIE	AM	AL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/(W)*	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]				DL	DS	TB	TS[1:0]		IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Descriptions
31	TC	0	R/W	Transfer Count Mode Specifies whether to transmit data once or for the count specified in DMATCR by one transfer request. Note that when this bit is set to 0, the TB bit must not be set to 1 (burst mode). When the USB, RSPI, SCIF_3, or IIC3 is selected for the transfer request source, this bit (TC) must not be set to 1. 0: Transmits data once by one transfer request 1: Transmits data for the count specified in DMATCR by one transfer request
30, 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
28	RLD	0	R/W	Reload Function Enable or Disable Enables or disables the reload function. 0: Disables the reload function 1: Enables the reload function
27 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 and CHCR_7; it is always read as 0 and the write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	TL	0	R/W	Transfer End Level Specifies the TEND signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is reserved in CHCR_2 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: Low-active output from TEND 1: High-active output from TEND
21, 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	Half-End Flag This bit is set to 1 when the transfer count reaches half of the DMATCR value that was specified before transfer starts. If DMA transfer ends because of an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR before the transfer count reaches half of the initial DMATCR value, the HE bit is not set to 1. If DMA transfer ends due to an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR after the HE bit is set to 1, the bit remains set to 1. To clear the HE bit, write 0 to it after HE = 1 is read. 0: DMATCR > (DMATCR set before transfer starts)/2 during DMA transfer or after DMA transfer is terminated [Clearing condition] • Writing 0 after reading HE = 1. 1: DMATCR ≤ (DMATCR set before transfer starts)/2
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether to issue an interrupt request to the CPU when the transfer count reaches half of the DMATCR value that was specified before transfer starts. When the HIE bit is set to 1, the DMAC requests an interrupt to the CPU when the HE bit becomes 1. 0: Disables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2 1: Enables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2

Bit	Bit Name	Initial Value	R/W	Descriptions
17	AM	0	R/W	Acknowledge Mode Specifies whether DACK is output in data read cycle or in data write cycle in dual address mode. In single address mode, DACK is always output regardless of the specification by this bit. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: DACK output in read cycle (dual address mode) 1: DACK output in write cycle (dual address mode)
16	AL	0	R/W	Acknowledge Level Specifies the DACK (acknowledge) signal output is high active or low active. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: Low-active output from DACK 1: High-active output from DACK Note: To use the DACK pins as high-active output, pull them down and perform the following settings. 1. After the reset start, specify the high-active output by this bit in CHCR for the DACK pins. 2. Then specify the DACK pins for the pin function controller setting. 3. The DACK pin setting in CHCR should be retained hereafter.

Bit	Bit Name	Initial Value	R/W	Descriptions
15, 14	DM[1:0]	00	R/W	Destination Address Mode These bits select whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, DM1 and DM0 bits are ignored when data is transferred to an external device with DACK.) 00: Fixed destination address (Setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer, setting prohibited in 16-byte transfer) 11: Setting prohibited
13, 12	SM[1:0]	00	R/W	Source Address Mode These bits select whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with DACK.) 00: Fixed source address (Setting prohibited in 16-byte-unit transfer) 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer, setting prohibited in 16-byte-unit transfer) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
11 to 8	RS[3:0]	0000	R/W	Resource Select These bits specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state when DMA enable bit (DE) is set to 0. 0000: External request, dual address mode 0001: Setting prohibited 0010: External request/single address mode External address space → External device with DACK 0011: External request/single address mode External device with DACK → External address space 0100: Auto request 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: DMA extension resource selector 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited Note: External request specification is valid only in CHCR_0 to CHCR_3. If a request source is selected in channels CHCR_4 to CHCR_7, no operation will be performed.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	DREQ Level
6	DS	0	R/W	DREQ Edge Select These bits specify the sampling method of the DREQ pin input and the sampling level. These bits are valid only in CHCR_0 to CHCR_3. These bits are reserved in CHCR_4 to CHCR_7; they are always read as 0 and the write value should always be 0. If the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, the specification by these bits is ignored. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies bus mode when DMA transfers data. Note that burst mode must not be selected when TC = 0. 0: Cycle steal mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	Transfer Size These bits specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte unit 01: Word unit (two bytes) 10: Longword unit (four bytes) 11: 16-byte unit (four longwords)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when TE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag This bit is set to 1 when DMATCR becomes 0 and DMA transfer ends. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR becomes 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR). To clear the TE bit, write 0 after reading TE = 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been terminated [Clearing condition] • Writing 0 after reading TE = 1 1: DMA transfer ends by the specified count (DMATCR = 0)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto-request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this case, all of the bits TE, NMIF in DMAOR, and AE must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0 as in the case of auto-request mode. Clearing the DE bit to 0 can terminate the DMA transfer. Before modifying the CHCR setting, clear the DE bit to 0 for the corresponding channel. 0: DMA transfer disabled 1: DMA transfer enabled

Note: * Only 0 can be written to clear the flag after 1 is read.

10.3.5 DMA Reload Source Address Registers (RSAR)

The DMA reload source address registers (RSAR) are 32-bit readable/writable registers.

When the reload function is enabled, the RSAR value is written to the source address register (SAR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RSAR during the current DMA transfer. When the reload function is disabled, RSAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

RSAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.6 DMA Reload Destination Address Registers (RDAR)

The DMA reload destination address registers (RDAR) are 32-bit readable/writable registers.

When the reload function is enabled, the RDAR value is written to the destination address register (DAR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RDAR during the current DMA transfer. When the reload function is disabled, RDAR is ignored.

To transfer data of 16-bit or 32-bit width, specify the address with 16-bit or 32-bit address boundary respectively. To transfer data in units of 16 bytes, set a value at a 16-byte boundary.

RDAR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.7 DMA Reload Transfer Count Registers (RDMATCR)

The DMA reload transfer count registers (RDMATCR) are 32-bit readable/writable registers.

When the reload function is enabled, the RDMATCR value is written to the transfer count register (DMATCR) at the end of the current DMA transfer. In this case, a new value for the next DMA transfer can be preset in RDMATCR during the current DMA transfer. When the reload function is disabled, RDMATCR is ignored.

The upper eight bits of RDMATCR are always read as 0, and the write value should always be 0.

As in DMATCR, the transfer count is 1 when the setting is H'00000001, 16,777,215 when H'00FFFFFF is set, and 16,777,216 (the maximum) when H'00000000 is set. To transfer data in 16 bytes, one 16-byte transfer (128 bits) counts one.

RDMATCR is initialized to H'00000000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.8 DMA Operation Register (DMAOR)

The DMA operation register (DMAOR) is a 16-bit readable/writable register that specifies the priority level of channels at the DMA transfer. This register also shows the DMA transfer status.

DMAOR is initialized to H'0000 by a reset and retains the value in software standby mode and module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CMS[1:0]		-	-	PR[1:0]		-	-	-	-	-	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	CMS[1:0]	00	R/W	Cycle Steal Mode Select These bits select either normal mode or intermittent mode in cycle steal mode. It is necessary that the bus modes of all channels be set to cycle steal mode to make intermittent mode valid. 00: Normal mode 01: Setting prohibited 10: Intermittent mode 16 Executes one DMA transfer for every 16 cycles of Bϕ clock. 11: Intermittent mode 64 Executes one DMA transfer for every 64 cycles of Bϕ clock.
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PR[1:0]	00	R/W	Priority Mode These bits select the priority level between channels when there are transfer requests for multiple channels simultaneously. 00: Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 01: Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7 10: Setting prohibited 11: Round-robin mode (only supported in CH0 to CH3)
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	AE	0	R/(W)*	Address Error Flag Indicates whether an address error has occurred by the DMAC. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading 1. 0: No DMAC address error 1: DMAC address error occurred [Clearing condition] • Writing 0 after having read this bit as 1. Write 1 after having read this bit as 0.
1	NMIF	0	R/(W)*	NMI Flag Indicates that an NMI interrupt occurred. When this bit is set, even if the DE bit in CHCR and the DME bit in DMAOR are set to 1, DMA transfer is not enabled. This bit can only be cleared by writing 0 after reading 1. When the NMI is input, the DMA transfer in progress can be done in one transfer unit. Even if the NMI interrupt is input while the DMAC is not in operation, the NMIF bit is set to 1. 0: No NMI interrupt 1: NMI interrupt occurred [Clearing condition] • Writing 0 after having read this bit as 1.

Bit	Bit Name	Initial Value	R/W	Description
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfer on all channels. If the DME bit and DE bit in CHCR are set to 1, DMA transfer is enabled. However, transfer is enabled only when the TE bit in CHCR of the transfer corresponding channel, the NMIF bit in DMAOR, and the AE bit are all cleared to 0. Clearing the DME bit to 0 can terminate the DMA transfer on all channels. 0: DMA transfer is disabled on all channels 1: DMA transfer is enabled on all channels

Note: * To clear the flag, write 0 after having read the flag as 1. Write 1 after having read the flag as 0. Only 0 can be written after 1 is read.

If the priority mode bits are modified after a DMA transfer, the channel priority is initialized. If fixed mode 2 is specified, the channel priority is specified as CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7. If round-robin mode is specified, the transfer end channel is reset.

Table 10.3 show the priority change in each mode (modes 0 to 2) specified by the priority mode bits. In each priority mode, the channel priority to accept the next transfer request may change in up to three ways according to the transfer end channel.

For example, when the transfer end channel is channel 1, the priority of the channel to accept the next transfer request is specified as CH2 > CH3 > CH0 >CH1 > CH4 > CH5 > CH6 > CH7. When the transfer end channel is any one of the channels 4 to 7, round-robin will not be applied and the priority level is not changed at the end of transfer in the channels 4 to 7.

The DMAC internal operation for an address error is as follows:

- No address error: Read (source to DMAC) → Write (DMAC to destination)
- Address error in source address: Nop → Nop
- Address error in destination address: Read → Nop

Table 10.3 Combinations of Priority Mode Bits

	Transfer End	Priority Mode Bits		Priority Level at the End of Transfer							
				High	◄						Low
Mode	**CH No.**	**PR[1]**	**PR[0]**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Mode 0 (fixed mode 1)	Any channel	0	0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
Mode 1 (fixed mode 2)	Any channel	0	1	CH0	CH4	CH1	CH5	CH2	CH6	CH3	CH7
Mode 2 (round-robin mode)	CH0	1	1	CH1	CH2	CH3	CH0	CH4	CH5	CH6	CH7
	CH1	1	1	CH2	CH3	CH0	CH1	CH4	CH5	CH6	CH7
	CH2	1	1	CH3	CH0	CH1	CH2	CH4	CH5	CH6	CH7
	CH3	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH4	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH5	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH6	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH7	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7

10.3.9 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3)

The DMA extension resource selectors (DMARS) are 16-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 is for channels 0 and 1, DMARS1 is for channels 2 and 3, DMARS2 is for channels 4 and 5, and DMARS3 is for channels 6 and 7. Table 10.4 shows the specifiable combinations.

DMARS can specify transfer requests from two USB sources, one RCAN source, two SSU sources, two SCIF sources, two IIC3 sources, one A/D converter source, five MTU2 sources, two CMT sources, four USB sources, one RCAN-ET source, and two RSPI sources.

DMARS is initialized to H'0000 by a reset and retains the value in software standby mode and module standby mode.

- DMARS0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH1 MID[5:0]						CH1 RID[1:0]		CH0 MID[5:0]						CH0 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH3 MID[5:0]						CH3 RID[1:0]		CH2 MID[5:0]						CH2 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH5 MID[5:0]						CH5 RID[1:0]		CH4 MID[5:0]						CH4 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DMARS3

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH7 MID[5:0]						CH7 RID[1:0]		CH6 MID[5:0]						CH6 RID[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer requests from the various modules specify MID and RID as shown in table 10.4.

Table 10.4 DMARS Settings

Peripheral Module		Setting Value for One Channel ({MID, RID})	MID	RID	Function
USB	USBTXI0	H'81	B'100000	B'01	Transmit
	USBRXI0	H'82		B'10	Receive
RCAN-ET	RM0_0	H'86	B'100001	B'10	Receive
RSPI	SPTI	H'89	B'100010	B'01	Transmit
	SPRI	H'8A		B'10	Receive
SCIF_3	TXI3	H'8D	B'100011	B'01	Transmit
	RXI3	H'8E		B'10	Receive
USB	USBTXI1	H'91	B'100100	B'01	Transmit
	USBRXI1	H'92		B'10	Receive
IIC3	TXI	H'A1	B'101000	B'01	Transmit
	RXI	H'A2		B'10	Receive
A/D converter_0	ADI0	H'B3	B'101100	B'11	—
MTU2_0	TGIA_0	H'E3	B'111000	B'11	—
MTU2_1	TGIA_1	H'E7	B'111001	B'11	—
MTU2_2	TGIA_2	H'EB	B'111010	B'11	—
MTU2_3	TGIA_3	H'EF	B'111011	B'11	—
MTU2_4	TGIA_4	H'F3	B'111100	B'11	—
CMT_0	CMI0	H'FB	B'111110	B'11	—
CMT_1	CMI1	H'FF	B'111111	B'11	—

When MID or RID other than the values listed in table 10.4 is set, the operation of this LSI is not guaranteed. The transfer request from DMARS is valid only when the resource select bits (RS[3:0]) in CHCR0 to CHCR7 have been set to B'1000. Otherwise, even if DMARS has been set, the transfer request source is not accepted.

10.4 Operation

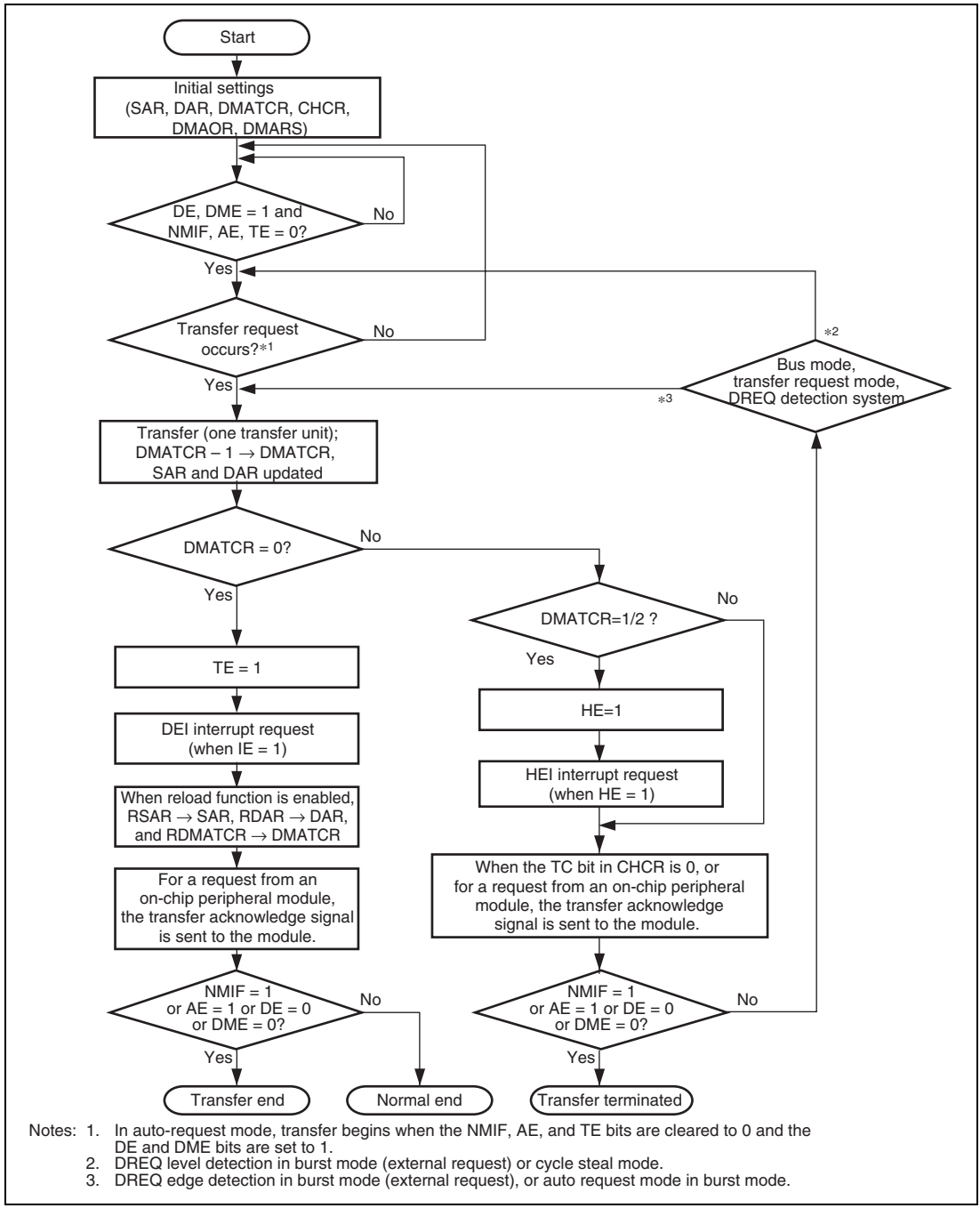
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. In bus mode, burst mode or cycle steal mode can be selected.

10.4.1 Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count registers (DMATCR), DMA channel control registers (CHCR), DMA operation register (DMAOR), and DMA extension resource selector (DMARS) are set for the target transfer conditions, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0)
2. When a transfer request comes and transfer is enabled, the DMAC transfers one transfer unit of data (depending on the TS0 and TS1 settings). For an auto request, the transfer begins automatically when the DE bit and DME bit are set to 1. The DMATCR value will be decremented by 1 for each transfer. The actual transfer flows vary by address mode and bus mode.
3. When half of the specified transfer count is exceeded (when DMATCR reaches half of the initial value), an HEI interrupt is sent to the CPU if the HIE bit in CHCR is set to 1.
4. When transfer has been completed for the specified count (when DMATCR reaches 0), the transfer ends normally. If the IE bit in CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
5. When an address error in the DMAC or an NMI interrupt is generated, the transfer is terminated. Transfers are also terminated when the DE bit in CHCR or the DME bit in DMAOR is cleared to 0.

Figure 10.2 is a flowchart of this procedure.



Notes: 1. In auto-request mode, transfer begins when the NMIF, AE, and TE bits are cleared to 0 and the DE and DME bits are set to 1.
2. DREQ level detection in burst mode (external request) or cycle steal mode.
3. DREQ edge detection in burst mode (external request), or auto request mode in burst mode.

Figure 10.2 DMA Transfer Flowchart

10.4.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated in external devices and on-chip peripheral modules that are neither the transfer source nor destination.

Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. The request mode is selected by the RS[3:0] bits in CHCR_0 to CHCR_7 and DMARS0 to DMARS3.

(1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the DE bits in CHCR_0 to CHCR_7 and the DME bit in DMAOR are set to 1, the transfer begins so long as the TE bits in CHCR_0 to CHCR_7, and the AE and NMIF bits in DMAOR are 0.

(2) External Request Mode

In this mode a transfer is performed at the request signals (DREQ0 to DREQ3) of an external device. Choose one of the modes shown in table 10.5 according to the application system. When the DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, NMIF = 0), DMA transfer is performed upon a request at the DREQ input.

Table 10.5 Selecting External Request Modes with the RS Bits

RS[3]	RS[2]	RS[1]	RS[0]	Address Mode	Transfer Source	Transfer Destination
0	0	0	0	Dual address mode	Any	Any
0	0	1	0	Single address mode	External memory, memory-mapped external device	External device with DACK
			1		External device with DACK	External memory, memory-mapped external device

Choose to detect DREQ by either the edge or level of the signal input with the DL and DS bits in CHCR_0 to CHCR_3 as shown in table 10.6. The source of the transfer request does not have to be the data transfer source or destination.

Table 10.6 Selecting External Request Detection with DL and DS Bits

CHCR		
DL bit	**DS bit**	**Detection of External Request**
0	0	Low level detection
	1	Falling edge detection
1	0	High level detection
	1	Rising edge detection

When DREQ is accepted, the DREQ pin enters the request accept disabled state (non-sensitive period). After issuing acknowledge DACK signal for the accepted DREQ, the DREQ pin again enters the request accept enabled state.

When DREQ is used by level detection, there are following two cases by the timing to detect the next DREQ after outputting DACK.

Overrun 0: Transfer is terminated after the same number of transfer has been performed as requests.

Overrun 1: Transfer is terminated after transfers have been performed for (the number of requests plus 1) times.

The DO bit in CHCR selects this overrun 0 or overrun 1.

Table 10.7 Selecting External Request Detection with DO Bit

CHCR	
DO bit	**External Request**
0	Overrun 0
1	Overrun 1

(3) On-Chip Peripheral Module Request

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

DMA transfer request signals from on-chip peripheral modules to the DMAC include transmit data empty and receive data full requests from the SCIF, A/D conversion end request from the A/D converter, compare match request from the CMT, and data transfer requests from the IIC3, MTU2, USB, RCAN-ET, and RSPI.

When a transfer request signal is sent in on-chip peripheral module request mode while DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, and NMIF = 0), DMA transfer is performed.

When the transmit data empty from the SCIF is selected, specify the transfer destination as the corresponding SCIF transmit data register. Likewise, when the receive data full from the SCIF is selected, specify the transfer source as the corresponding SCIF receive data register. When a transfer request is made by the A/D converter, the transfer source must be the A/D data register (ADDR). When the IIC3 transmission is selected as the transfer request, the transfer destination must be ICDRT; when the IIC3 reception is selected as the transfer request, the transfer source must be ICDRR. When the USB transmission is selected as the transfer request, the transfer destination must be the data registers (USBEPDR2 and USBEPDR5) for the corresponding endpoint; when the USB reception is selected as the transfer request, the transfer source must be the data registers (USBEPDR1 and USBEPDR4) for the corresponding endpoint. When the RSPI transmission is selected as the transfer request, the transfer destination must be the RSPI data register (SPDR); when the RSPI reception is selected as the transfer request, the transfer source must be the RSPI data register (SPDR). When the RCAN-ET receive interrupt is selected as the transfer request, the transfer source must be a mailbox (MB0 to MB15). Any address can be specified for data transfer source and destination when a transfer request is sent from the CMT or MTU2.

Table 10.8 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer	DMA Transfer	Transfer	Transfer	Bus
RS[3:0]	MID	RID	Request Source	Request Signal	Source	Destination	Mode
1000	100000	01	USB transmit	EP2 FIFO empty transfer request	Any	USBEPDR2	Cycle steal
		10	USB receive	EP1 FIFO full transfer request	USBEPDR1	Any	
	100001	10	RCAN-ET	RM0 (RCAN-ET receive interrupt)	MB0 to MB15*1	Any	Cycle steal
	100010	01	RSPI transmit	SPTI (transmit data empty)	Any	SPDR	Cycle steal or burst*2
		10	RSPI receive	SPRI (receive data full)	SPDR	Any	
	100011	01	SCIF_3 transmit	TXI3 (transmit FIFO data empty)	Any	SCFTDR3	Cycle steal
		10	SCIF_3 receive	RXI3 (receive FIFO data full)	SCFRDR3	Any	
	100100	01	USB transmit	EP5 FIFO empty transfer request	Any	USBEPDR5	Cycle steal
		10	USB receive	EP4 FIFO full transfer request	USBEPDR4	Any	
	101000	01	IIC3 transmit	TXI (transmit data empty)	Any	ICDRT	Cycle steal
		10	IIC3 receive	RXI (receive data full)	ICDRR	Any	
	101100	11	A/D converter_0	ADI0 (A/D conversion end)	ADDR0 to ADDR3	Any	Cycle steal
	111000	11	MTU2_0	TGIA_0	Any	Any	Cycle steal or burst
	111001	11	MTU2_1	TGIA_1	Any	Any	
	111010	11	MTU2_2	TGIA_2	Any	Any	
	111011	11	MTU2_3	TGIA_3	Any	Any	
	111100	11	MTU2_4	TGIA_4	Any	Any	
	111110	11	CMT_0	Compare match transmit request 0	Any	Any	Cycle steal or burst
	111111	11	CMT_1	Compare match transmit request 1	Any	Any	

Notes: 1. Transfer count mode can be used to read message control fields 1 to 2 in a mailbox.
2. To set to burst mode, see section 18.5.2, DMAC Burst Transfer.

10.4.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Three modes (fixed mode 1, fixed mode 2, and round-robin mode) are selected using the PR1 and PR0 bits in DMAOR.

(1) Fixed Mode

In fixed modes, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7

Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7

These are selected by the PR1 and PR0 bits in the DMA operation register (DMAOR).

(2) Round-Robin Mode

Each time one unit of word, byte, longword, or 16 bytes is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished is rotated to the lowest of the priority order among the four round-robin channels (channels 0 to 4). The priority of the channels other than the round-robin channels (channels 0 to 4) does not change even in round-robin mode. The round-robin mode operation is shown in figure 10.3. The priority in round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 immediately after a reset.

When round-robin mode has been specified, do not concurrently specify cycle steal mode and burst mode as the bus modes of any two or more channels.

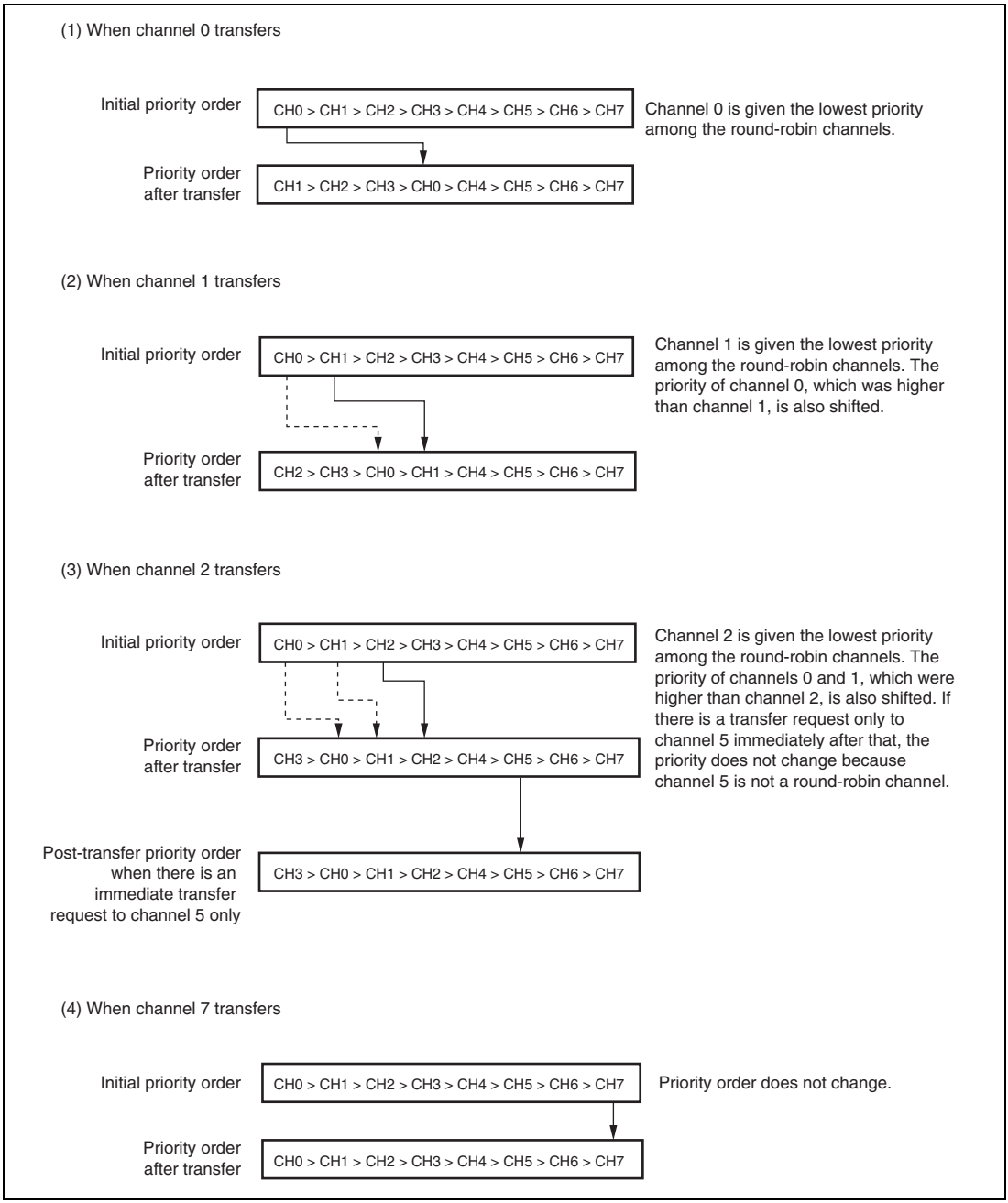


Figure 10.3 Round-Robin Mode

Figure 10.4 shows how the priority order changes when channel 0 and channel 3 transfers are requested simultaneously and a channel 1 transfer is requested during the channel 0 transfer. The DMAC operates as follows:

1. Transfer requests are generated simultaneously to channels 0 and 3.
2. Channel 0 has a higher priority, so the channel 0 transfer begins first (channel 3 waits for transfer).
3. A channel 1 transfer request occurs during the channel 0 transfer (channels 1 and 3 are both waiting)
4. When the channel 0 transfer ends, channel 0 is given the lowest priority among the round-robin channels.
5. At this point, channel 1 has a higher priority than channel 3, so the channel 1 transfer begins (channel 3 waits for transfer).
6. When the channel 1 transfer ends, channel 1 is given the lowest priority among the round-robin channels.
7. The channel 3 transfer begins.
8. When the channel 3 transfer ends, channels 3 and 2 are lowered in priority so that channel 3 is given the lowest priority among the round-robin channels.

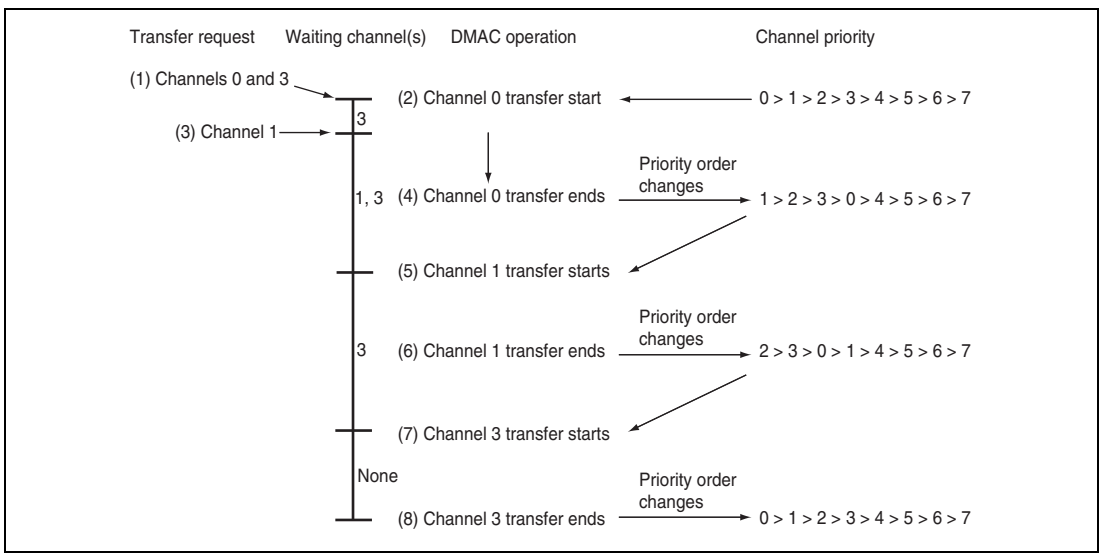


Figure 10.4 Changes in Channel Priority in Round-Robin Mode

10.4.4 DMA Transfer Types

DMA transfer has two types: single address mode transfer and dual address mode transfer. They depend on the number of bus cycles of access to the transfer source and destination. A data transfer timing depends on the bus mode, which is cycle steal mode or burst mode. The DMAC supports the transfers shown in table 10.9.

Table 10.9 Supported DMA Transfers

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Peripheral Module**	**On-Chip Memory**
External device with DACK	Not available	Dual, single	Dual, single	Not available	Not available
External memory	Dual, single	Dual	Dual	Dual	Dual
Memory-mapped external device	Dual, single	Dual	Dual	Dual	Dual
On-chip peripheral module	Not available	Dual	Dual	Dual	Dual
On-chip memory	Not available	Dual	Dual	Dual	Dual

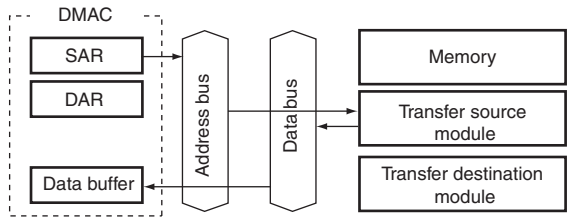
Notes: 1. Dual: Dual address mode
2. Single: Single address mode
3. 16-byte transfer is available only for on-chip peripheral modules that support longword access.

(1) Address Modes

(a) Dual Address Mode

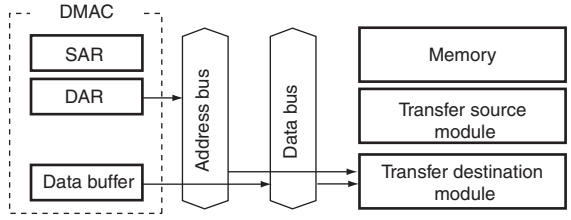
In dual address mode, both the transfer source and destination are accessed (selected) by an address. The transfer source and destination can be located externally or internally.

DMA transfer requires two bus cycles because data is read from the transfer source in a data read cycle and written to the transfer destination in a data write cycle. At this time, transfer data is temporarily stored in the DMAC. In the transfer between external memories as shown in figure 10.5, data is read to the DMAC from one external memory in a data read cycle, and then that data is written to the other external memory in a data write cycle.



The SAR value is an address, data is read from the transfer source module, and the data is tempolarily stored in the DMAC.

First bus cycle



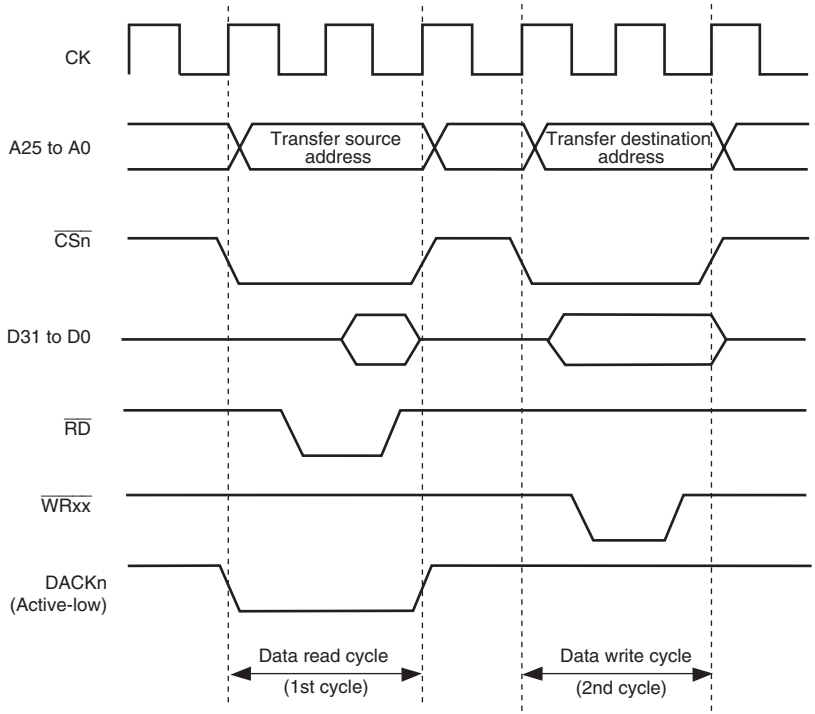
The DAR value is an address and the value stored in the data buffer in the DMAC is written to the transfer destination module.

Second bus cycle

Figure 10.5 Data Flow of Dual Address Mode

Auto request, external request, and on-chip peripheral module request are available for the transfer request. DACK can be output in read cycle or write cycle in dual address mode. The AM bit in the channel control register (CHCR) can specify whether the DACK is output in read cycle or write cycle.

Figure 10.6 shows an example of DMA transfer timing in dual address mode.



Note: In transfer between external memories, with DACK output in the read cycle, DACK output timing is the same as that of CSn.

Figure 10.6 Example of DMA Transfer Timing in Dual Mode (Transfer Source: Normal Memory, Transfer Destination: Normal Memory)

(b) Single Address Mode

In single address mode, both the transfer source and destination are external devices, either of them is accessed (selected) by the DACK signal, and the other device is accessed by an address. In this mode, the DMAC performs one DMA transfer in one bus cycle, accessing one of the external devices by outputting the DACK transfer request acknowledge signal to it, and at the same time outputting an address to the other device involved in the transfer. For example, in the case of transfer between external memory and an external device with DACK shown in figure 10.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

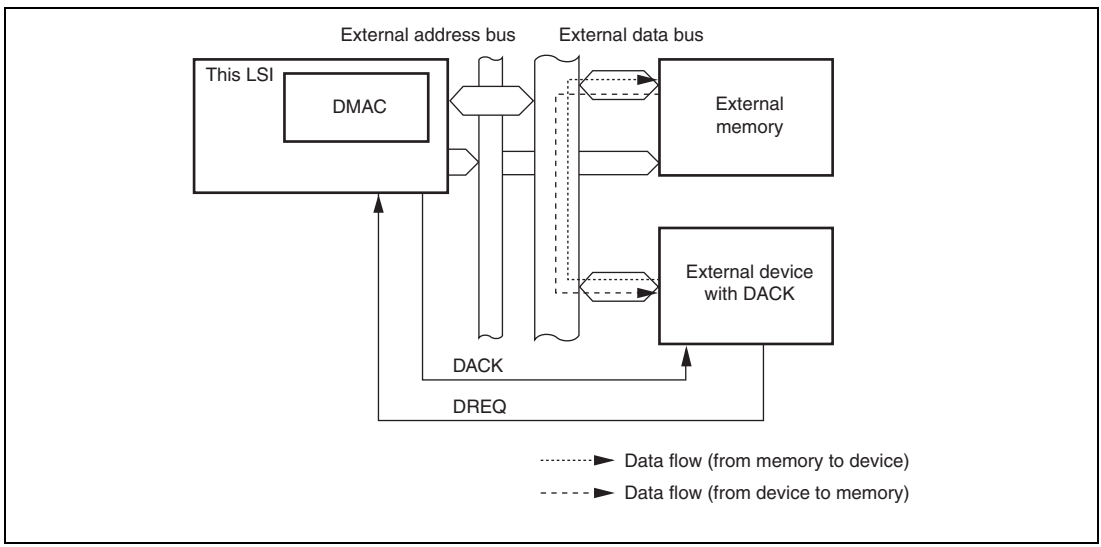
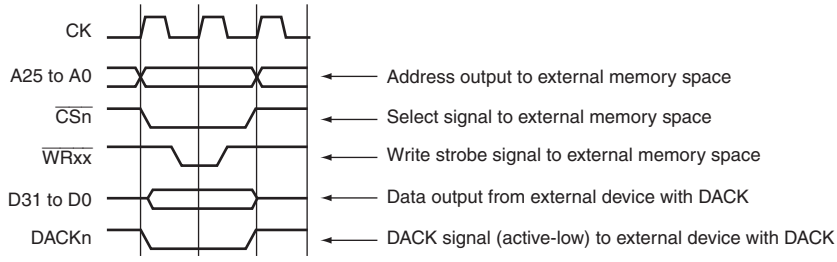


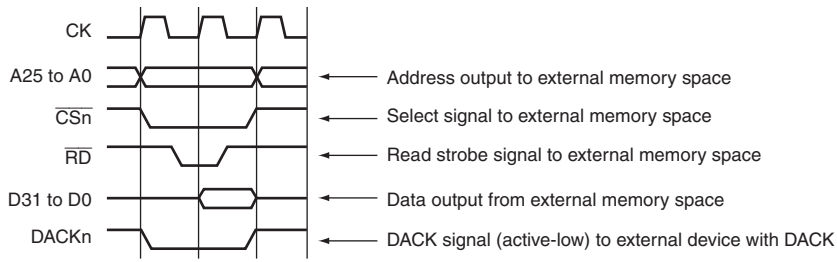
Figure 10.7 Data Flow in Single Address Mode

Two kinds of transfer are possible in single address mode: (1) transfer between an external device with DACK and a memory-mapped external device, and (2) transfer between an external device with DACK and external memory. In both cases, only the external request signal (DREQ) is used for transfer requests.

Figure 10.8 shows an example of DMA transfer timing in single address mode.



(a) External device with DACK → External memory space (normal memory)



(b) External memory space (normal memory) → External device with DACK

Figure 10.8 Example of DMA Transfer Timing in Single Address Mode

(2) Bus Modes

There are two bus modes; cycle steal and burst. Select the mode by the TB bits in the channel control registers (CHCR).

(a) Cycle Steal Mode

- Normal mode

  In normal mode of cycle steal, the bus mastership is given to another bus master after a one-transfer-unit (byte, word, longword, or 16-byte unit) DMA transfer. When another transfer request occurs, the bus mastership is obtained from another bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus mastership is passed to another bus master. This is repeated until the transfer end conditions are satisfied.

  The cycle-steal normal mode can be used for any transfer section; transfer request source, transfer source, and transfer destination.

  Figure 10.9 shows an example of DMA transfer timing in cycle-steal normal mode. Transfer conditions shown in the figure are:

— Dual address mode
— DREQ low level detection

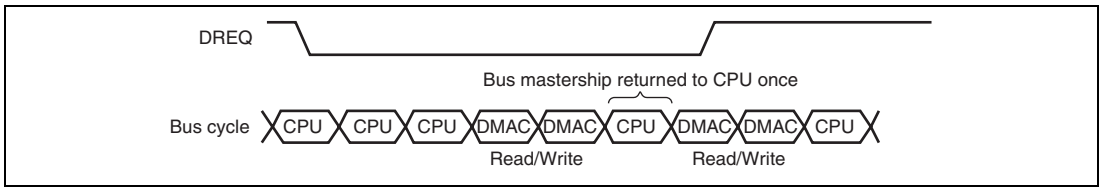


Figure 10.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- Intermittent Mode 16 and Intermittent Mode 64

  In intermittent mode of cycle steal, DMAC returns the bus mastership to other bus master whenever a unit of transfer (byte, word, longword, or 16 bytes) is completed. If the next transfer request occurs after that, DMAC obtains the bus mastership from other bus master after waiting for 16 or 64 cycles of Bϕ clock. DMAC then transfers data of one unit and returns the bus mastership to other bus master. These operations are repeated until the transfer end condition is satisfied. It is thus possible to make lower the ratio of bus occupation by DMA transfer than normal mode of cycle steal.

  The cycle-steal intermittent mode can be used for any transfer section; transfer request source, transfer source, and transfer destination. The bus modes, however, must be cycle steal mode in all channels.

  Figure 10.10 shows an example of DMA transfer timing in cycle-steal intermittent mode. Transfer conditions shown in the figure are:

  — Dual address mode
  — DREQ low level detection

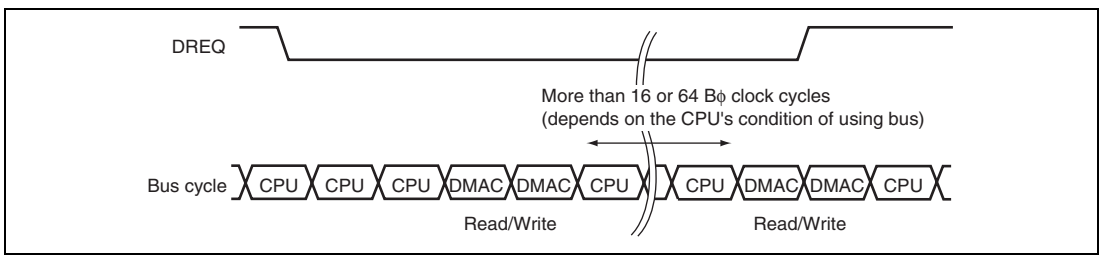


Figure 10.10 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(b) Burst Mode

In burst mode, once the DMAC obtains the bus mastership, it does not release the bus mastership and continues to perform transfer until the transfer end condition is satisfied. In external request mode with low level detection of the DREQ pin, however, when the DREQ pin is driven high, the bus mastership is passed to another bus master after the DMAC transfer request that has already been accepted ends, even if the transfer end conditions have not been satisfied.

Figure 10.11 shows DMA transfer timing in burst mode.

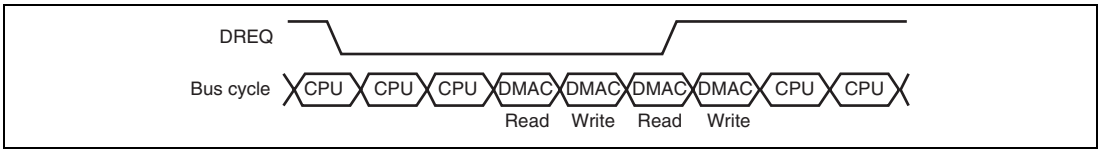


Figure 10.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 10.10 shows the relationship between request modes and bus modes by DMA transfer category.

Table 10.10 Relationship of Request Modes and Bus Modes by DMA Transfer Category

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External device with DACK and external memory	External	B/C	8/16/32/128	0 to 3
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0 to 3
	External memory and external memory	All*4	B/C	8/16/32/128	0 to 7*3
	External memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 7*3
	Memory-mapped external device and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 7*3
	External memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	Memory-mapped external device and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	On-chip peripheral module and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	On-chip memory and on-chip memory	All*4	B/C	8/16/32/128	0 to 7*3
	On-chip memory and memory-mapped external device	All*4	B/C	8/16/32/128	0 to 7*3
	On-chip memory and on-chip peripheral module	All*1	B/C*5	8/16/32/128*2	0 to 7*3
	On-chip memory and external memory	All*4	B/C	8/16/32/128	0 to 7*3
Single	External device with DACK and external memory	External	B/C	8/16/32/128	0 to 3
	External device with DACK and memory-mapped external device	External	B/C	8/16/32/128	0 to 3

[Legend]

B: Burst

C: Cycle steal

Notes: 1. External requests, auto requests, and on-chip peripheral module requests are all available. However, along with the exception of CMT and MTU2 as the transfer request source, the requesting module must be designated as the transfer source or the transfer destination.

2. Access size permitted for the on-chip peripheral module register functioning as the transfer source or transfer destination.

3. If the transfer request is an external request, channels 0 to 3 are only available.

4. External requests, auto requests, and on-chip peripheral module requests are all available. In the case of on-chip peripheral module requests, however, the CMT and MTU2 are only available.

5. Only cycle steal except for the RSPI, MTU2, and CMT as the transfer request source.

(4) Bus Mode and Channel Priority

In priority fixed mode (CH0 > CH1), when channel 1 is transferring data in burst mode and a request arrives for transfer on channel 0, which has higher-priority, the data transfer on channel 0 will begin immediately. In this case, if the transfer on channel 0 is also in burst mode, the transfer on channel 1 will only resume on completion of the transfer on channel 0.

When channel 0 is in cycle steal mode, one transfer-unit of data on this channel, which has the higher priority, is transferred. Data is then transferred continuously to channel 1 without releasing the bus. The bus mastership will then switch between the two in this order: channel 0, channel 1, channel 0, channel 1, etc. That is, the CPU cycle after the data transfer in cycle steal mode is replaced with a burst-mode transfer cycle (priority execution of burst-mode cycle). An example of this is shown in figure 10.12.

When multiple channels are in burst mode, data transfer on the channel that has the highest priority is given precedence. When DMA transfer is being performed on multiple channels, the bus mastership is not released to another bus-master device until all of the competing burst-mode transfers have been completed.

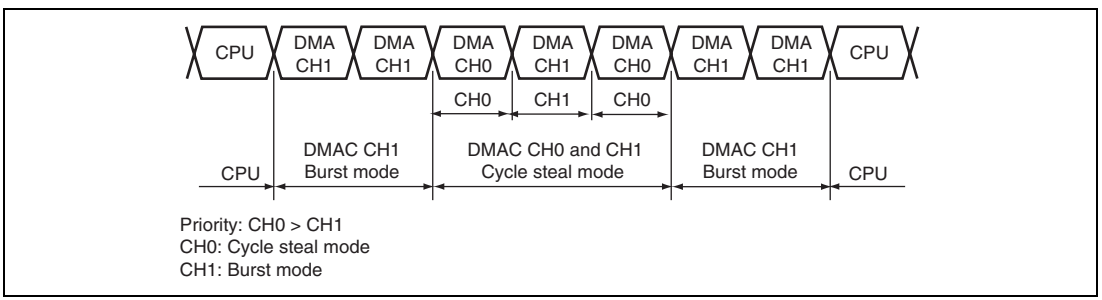


Figure 10.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 10.3. Note that channels in cycle steal and burst modes must not be mixed.

10.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing

(1) Number of Bus Cycles

When the DMAC is the bus master, the number of bus cycles is controlled by the bus state controller (BSC) in the same way as when the CPU is the bus master. For details, see section 9, Bus State Controller (BSC).

(2) DREQ Pin Sampling Timing

Figures 10.13 to 10.16 show the DREQ input sampling timings in each bus mode.

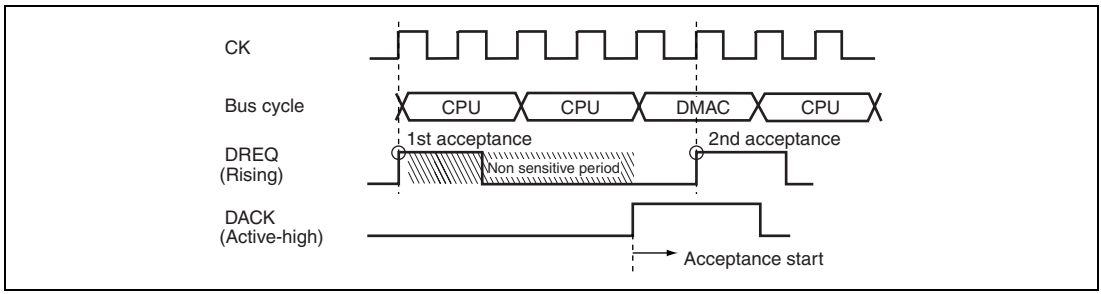


Figure 10.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

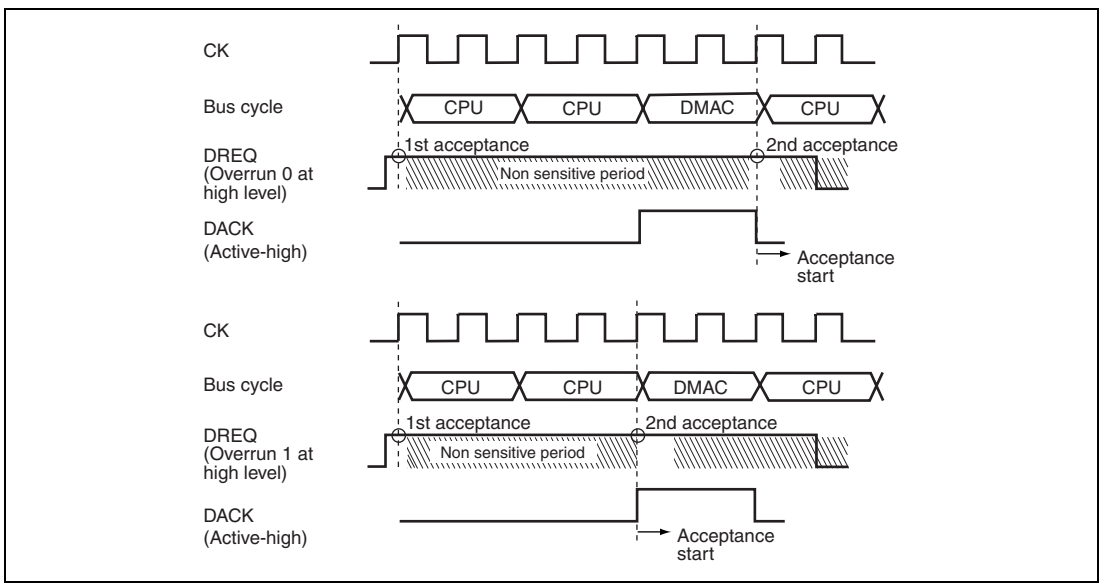


Figure 10.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

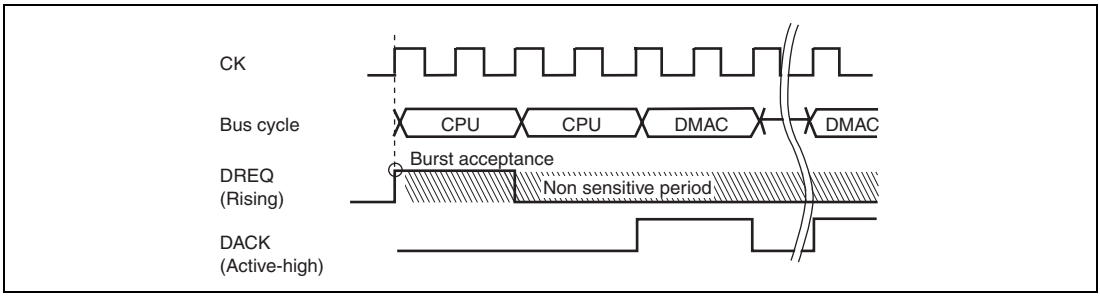


Figure 10.15 Example of DREQ Input Detection in Burst Mode Edge Detection

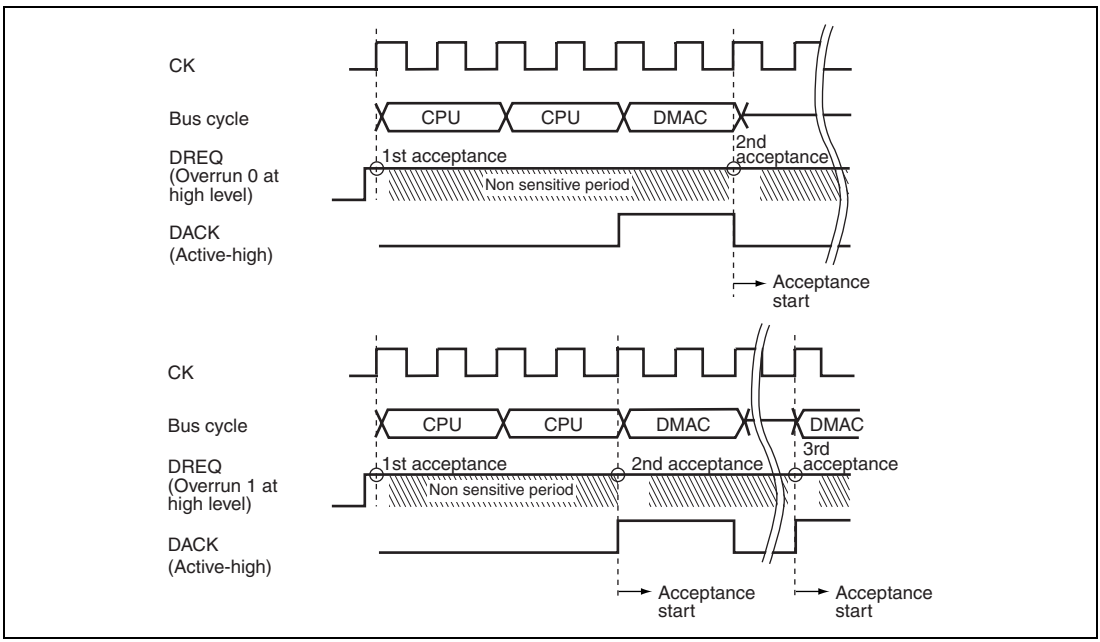


Figure 10.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 10.17 shows the TEND output timing.

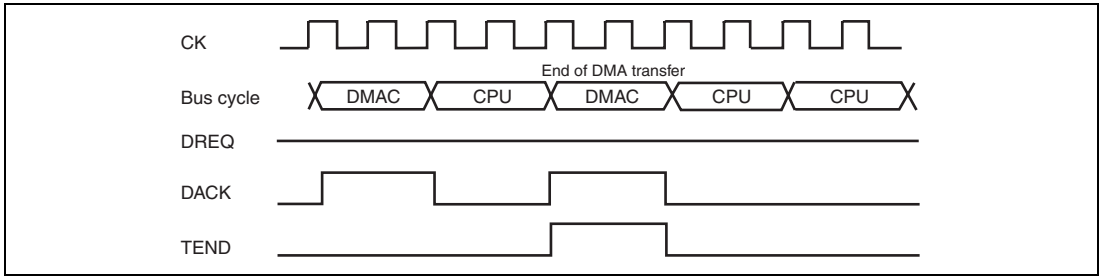
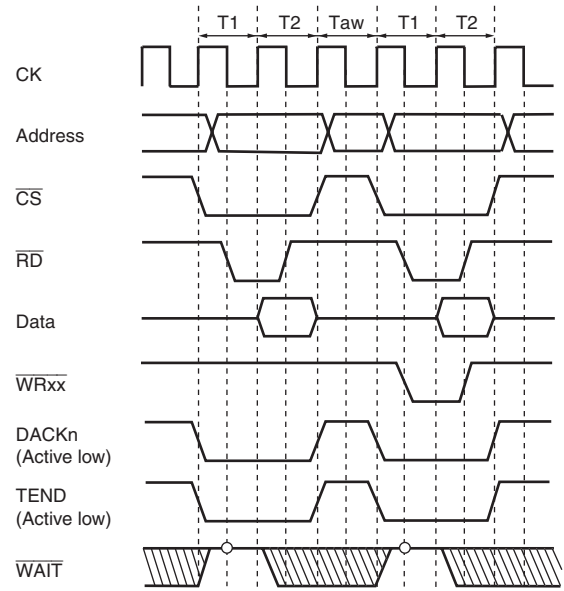


Figure 10.17 Example of DMA Transfer End Signal Timing (Cycle Steal Mode Level Detection)

The unit of the DMA transfer is divided into multiple bus cycles when 16-byte transfer is performed for an 8-bit or 16-bit external device, when longword access is performed for an 8-bit or 16-bit external device, or when word access is performed for an 8-bit external device. When a setting is made so that the DMA transfer size is divided into multiple bus cycles and the $\overline{CS}$ signal is negated between bus cycles, note that DACK and TEND are divided like the $\overline{CS}$ signal for data alignment. Also, if the DREQ detection is set to level-detection mode (DS bit in CHCR = 0), the DREQ sampling may not be detected correctly with divided DACK, and one extra overrun may occur at maximum.

Use a setting that does not divide DACK or specify a transfer size smaller than the external device bus width if DACK is divided. Figure 10.18 shows this example.



Note: TEND is asserted for the last unit of DMA transfer. If a transfer unit is divided into multiple bus cycles and the CS is negated between the bus cycles, TEND is also divided.

Figure 10.18 BSC Normal Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

10.5 Interrupt Sources

10.5.1 Interrupt Sources and Priority Order

The interrupt sources of the DMAC are the data transfer end interrupt (DEI) and data transfer half-end interrupt (HEI) for each channel. Table 10.11 lists the interrupt sources and their order of priority.

The IE and HIE bits in the DMA channel control registers (CHCRs) enable or disable the respective interrupt sources. Furthermore, the interrupt requests are independently conveyed to the interrupt controller.

A data-transfer end interrupt (DEI) is generated when, the transfer end flag and the transfer end interrupt enable (IE) bit in the DMA channel control register (CHCR) are set to 1. A data-transfer half end interrupt (HEI) is generated when the half-end flag and the half-end interrupt enable (HIE) bit in the DMA channel control register (CHCR) are set to 1. Clearing the interrupt flag bit to 0 cancels the interrupt request.

Priority among the channels is adjustable by the interrupt controller. The order of priority for interrupts of a given channel is fixed. For details, refer to section 6, Interrupt Controller (INTC).

Table 10.11 Interrupt Sources

Channel	Interrupt Source	Interrupt Enable Bit	Interrupt Flag	Priority
0	Data transfer end interrupt (DEI0)	IE	TE	High
	Data transfer half end interrupt (HEI0)	HIE	HE	
1	Data transfer end interrupt (DEI1)	IE	TE	
	Data transfer half end interrupt (HEI1)	HIE	HE	
2	Data transfer end interrupt (DEI2)	IE	TE	
	Data transfer half end interrupt (HEI2)	HIE	HE	
3	Data transfer end interrupt (DEI3)	IE	TE	
	Data transfer half end interrupt (HEI3)	HIE	HE	
4	Data transfer end interrupt (DEI4)	IE	TE	
	Data transfer half end interrupt (HEI4)	HIE	HE	
5	Data transfer end interrupt (DEI5)	IE	TE	
	Data transfer half end interrupt (HEI5)	HIE	HE	
6	Data transfer end interrupt (DEI6)	IE	TE	
	Data transfer half end interrupt (HEI6)	HIE	HE	
7	Data transfer end interrupt (DEI7)	IE	TE	
	Data transfer half end interrupt (HEI7)	HIE	HE	Low

10.6 Usage Notes

10.6.1 Setting of the Half-End Flag and the Half-End Interrupt

Since the following points for caution apply in cases where reference to the state of the half-end flag in the CHCR register or the half-end interrupt is used in conjunction with the reload function, please take care on these points.

Ensure that the reloaded number of transfers (the value set in RDMATCR) is always the same as the number of transfers that was initially set (the value set in DMATCR). If the initial setting in DMATCR and the value for the second and later transfers in RDMATCR are different, the timing with which the half-end flag is set may be faster than half the number of transfers, or the half-end flag might not be set at all. The same considerations apply to the half-end interrupt.

10.6.2 Timing of DACK and TEND Outputs

When the external memory is MPX-I/O or burst MPX-I/O, assertion of the DACK output has the same timing as the data cycle. For details, see the respective figures under section 9.5.5, MPX-I/O Interface, in section 9, Bus State Controller (BSC).

When the memory is other than the MPX-I/O or burst MPX-I/O, the DACK output is asserted with the same timing as the corresponding CS signal.

The TEND output does not depend on the type of memory and is always asserted with the same timing as the corresponding CS signal.

10.6.3 CHCR Setting

When changing the CHCR setting, the DE bit of the relevant channel must be cleared before the change.

10.6.4 Note on Activation of Multiple Channels

The same internal request must not be set to more than one channel.

10.6.5 Note on Transfer Request Input

A transfer request should be input after the DMAC settings have been made.

10.6.6 Conflict between NMI Interrupt and DMAC Activation

When a conflict occurs between the generation of the NMI interrupt and the DMAC activation, the NMI interrupt has priority. Thus the NMIF bit is set to 1 and the DMAC is not activated.

It takes 2 × Bcyc or 3 × Pcyc for checking DMAC stop by the NMI, 4 × Bcyc for checking DMAC activation by the DREQ, and 1 × Bcyc + 1 × Pcyc for checking DMAC activation by a peripheral module (Bcyc indicates the cycle of the external bus clock, Pcyc indicates the cycle of the peripheral clock).

10.6.7 Number of On-Chip RAM Access Cycles from DMAC

The number of on-chip RAM access cycles from the DMAC becomes the number of cycles shown in table 10.12, depending on whether the operation is read or write and the clock ratio between Iϕ (internal clock) and Bϕ (external bus clock).

Table 10.12 Number of On-Chip RAM Access Cycles from DMAC

Setting of Iϕ:Bϕ	Read Operation	Write Operation
1:1	3 × Bcyc	2 × Bcyc
1:1/2	2 × Bcyc	2 × Bcyc
1:1/4	2 × Bcyc	2 × Bcyc
Smaller than 1:1/4	1 × Bcyc	1 × Bcyc

Note: Bcyc indicates the cycle of the external bus clock.

Section 11 Multi-Function Timer Pulse Unit 2 (MTU2)

This LSI has an on-chip multi-function timer pulse unit 2 (MTU2) that comprises six 16-bit timer channels.

11.1 Features

- Maximum 16 pulse input/output lines and three pulse input lines
- Selection of eight counter input clocks for each channel (four clocks for channel 5)
- The following operations can be set for channels 0 to 4:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture is possible
  - Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 12-phase PWM output is possible in combination with synchronous operation.
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- Dead time compensation counter available in channel 5
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

Table 11.1 MTU2 Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		Pφ/1 Pφ/4 Pφ/16 Pφ/64 TCLKA TCLKB TCLKC TCLKD	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/1024 TCLKA TCLKB TCLKC	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 Pφ/1024 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64 Pφ/256 Pφ/1024 TCLKA TCLKB	Pφ/1 Pφ/4 Pφ/16 Pφ/64
General registers		TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRU_5 TGRV_5 TGRW_5
General registers/ buffer registers		TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4	—
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	Input pins TIC5U TIC5V TIC5W
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	√	√	√	√	√	—
	1 output	√	√	√	√	√	—
	Toggle output	√	√	√	√	√	—
Input capture function		√	√	√	√	√	√
Synchronous operation		√	√	√	√	√	—
PWM mode 1		√	√	√	√	√	—
PWM mode 2		√	√	√	—	—	—
Complementary PWM mode		—	—	—	√	√	—
Reset PWM mode		—	—	—	√	√	—
AC synchronous motor drive mode		√	—	—	√	√	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Phase counting mode	—	√	√	—	—	—
Buffer operation	√	—	—	√	√	—
Dead time compensation counter function	—	—	—	—	—	√
DMAC activation	TGRA_0 compare match or input capture	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture and TCNT overflow or underflow	—
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 or TCNT overflow or underflow	TGR compare match or input capture
A/D converter start trigger	TGRA_0 compare match or input capture TGRE_0 compare match	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture TCNT_4 underflow (trough) in complementary PWM mode	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Interrupt sources	7 sources	4 sources	4 sources	5 sources	5 sources	3 sources
	• Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	• Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	• Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	• 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	• 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	• Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA_4 and TCNT_4 • A/D converter start request at a match between TADCORB_4 and TCNT_4	—
Interrupt skipping function	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts	—

[Legend]

√: Possible

—: Not possible

Figure 11.1 shows a block diagram of the MTU2.

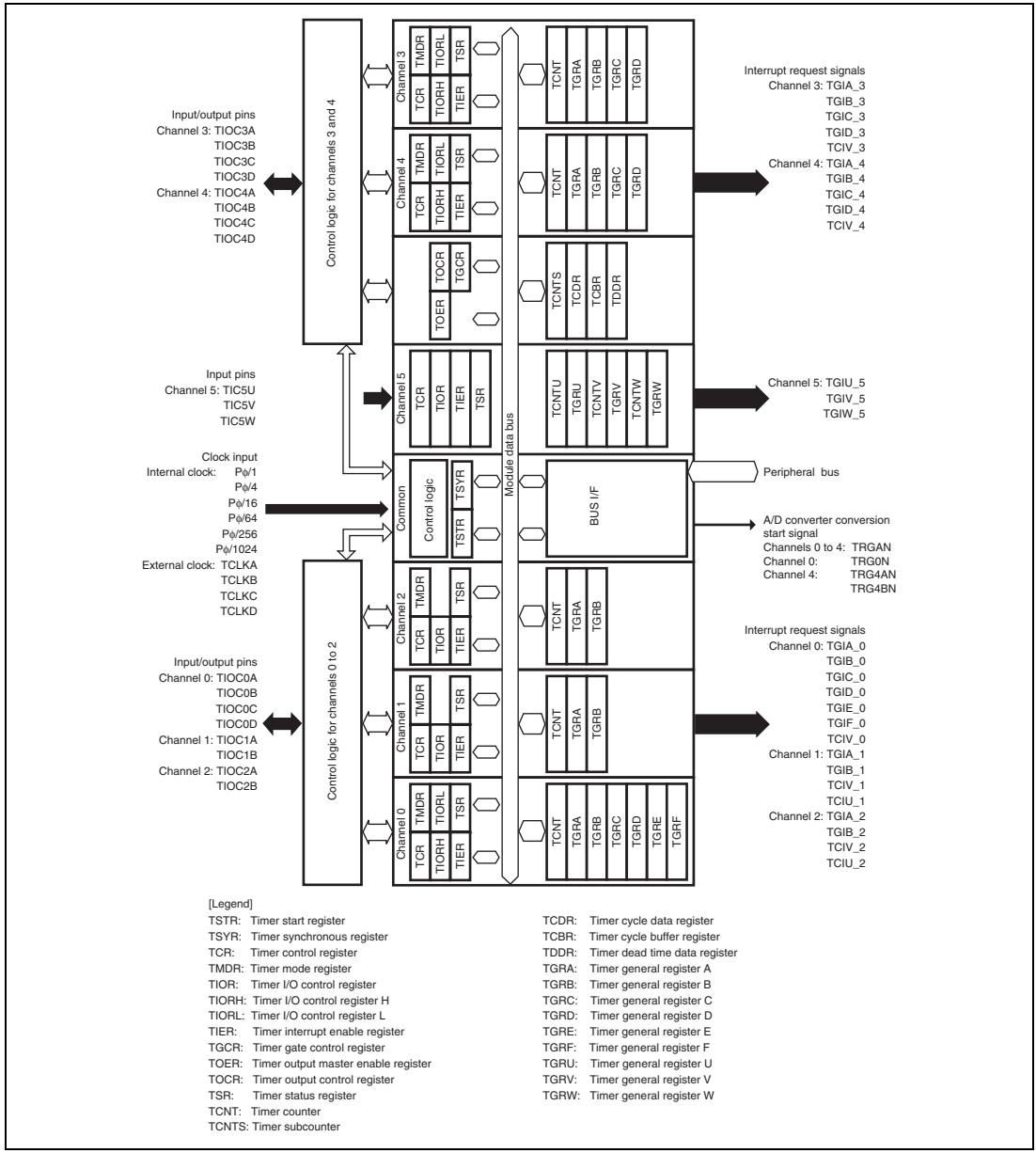


Figure 11.1 Block Diagram of MTU2

11.2 Input/Output Pins

Table 11.2 Pin Configuration

Channel	Pin Name	I/O	Function
Common	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	TIOC0A	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOC0B	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOC0C	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOC0D	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOC1A	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOC1B	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOC2A	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOC2B	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOC3A	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOC3B	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOC3C	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOC3D	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOC4A	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOC4B	I/O	TGRB_4 input capture input/output compare output/PWM output pin
	TIOC4C	I/O	TGRC_4 input capture input/output compare output/PWM output pin
	TIOC4D	I/O	TGRD_4 input capture input/output compare output/PWM output pin
5	TIC5U	Input	TGRU_5 input capture input/external pulse input pin
	TIC5V	Input	TGRV_5 input capture input/external pulse input pin
	TIC5W	Input	TGRW_5 input capture input/external pulse input pin

11.3 Register Descriptions

The MTU2 has the following registers. For details on register addresses and register states during each process, refer to section 32, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

Table 11.3 Register Descriptions

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200	8, 16, 32
Timer control register_4	TCR_4	R/W	H'00	H'FFFE4201	8
Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202	8, 16
Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE4203	8
Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204	8, 16, 32
Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE4206	8, 16
Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207	8
Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208	8, 16
Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE4209	8
Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A	8
Timer gate control register	TGCR	R/W	H'80	H'FFFE420D	8
Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E	8, 16
Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F	8
Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210	16, 32
Timer counter_4	TCNT_4	R/W	H'0000	H'FFFE4212	16
Timer cycle data register	TCDR	R/W	H'FFFF	H'FFFE4214	16, 32
Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216	16
Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16, 32
Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FFFE421C	16, 32

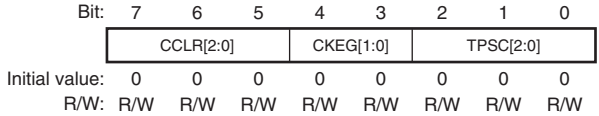
Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FFFE421E	16
Timer subcounter	TCNTS	R	H'0000	H'FFFE4220	16, 32
Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE4222	16
Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16, 32
Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FFFE4228	16, 32
Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	16
Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C	8, 16
Timer status register_4	TSR_4	R/W	H'C0	H'FFFE422D	8
Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE4230	8, 16
Timer interrupt skipping counter	TITCNT	R	H'00	H'FFFE4231	8
Timer buffer transfer set register	TBTER	R/W	H'00	H'FFFE4232	8
Timer dead time enable register	TDER	R/W	H'01	H'FFFE4234	8
Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236	8
Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238	8, 16
Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FFFE4239	8
Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FFFE4240	16
Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FFFE4244	16, 32
Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FFFE4246	16
Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE4248	16, 32
Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16
Timer waveform control register	TWCR	R/W	H'00	H'FFFE4260	8
Timer start register	TSTR	R/W	H'00	H'FFFE4280	8, 16
Timer synchronous register	TSYR	R/W	H'00	H'FFFE4281	8
Timer counter synchronous start register	TCSYSTR	R/W	H'00	H'FFFE4282	8

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284	8
Timer control register_0	TCR_0	R/W	H'00	H'FFFE4300	8, 16, 32
Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE4301	8
Timer I/O control registerH_0	TIORH_0	R/W	H'00	H'FFFE4302	8, 16
Timer I/O control registerL_0	TIORL_0	R/W	H'00	H'FFFE4303	8
Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE4304	8, 16, 32
Timer status register_0	TSR_0	R/W	H'C0	H'FFFE4305	8
Timer counter_0	TCNT_0	R/W	H'0000	H'FFFE4306	16
Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FFFE4308	16, 32
Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FFFE430A	16
Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FFFE430C	16, 32
Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FFFE430E	16
Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FFFE4320	16, 32
Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FFFE4322	16
Timer interrupt enable register2_0	TIER2_0	R/W	H'00	H'FFFE4324	8, 16
Timer status register2_0	TSR2_0	R/W	H'C0	H'FFFE4325	8
Timer buffer operation transfer mode register_0	TBTM_0	R/W	H'00	H'FFFE4326	8
Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380	8, 16
Timer mode register_1	TMDR_1	R/W	H'00	H'FFFE4381	8
Timer I/O control register_1	TIOR_1	R/W	H'00	H'FFFE4382	8
Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FFFE4384	8, 16, 32
Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385	8
Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE4386	16
Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16, 32

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FFFE438A	16
Timer input capture control register	TICCR	R/W	H'00	H'FFFE4390	8
Timer control register_2	TCR_2	R/W	H'00	H'FFFE4000	8, 16
Timer mode register_2	TMDR_2	R/W	H'00	H'FFFE4001	8
Timer I/O control register_2	TIOR_2	R/W	H'00	H'FFFE4002	8
Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FFFE4004	8, 16, 32
Timer status register_2	TSR_2	R/W	H'C0	H'FFFE4005	8
Timer counter_2	TCNT_2	R/W	H'0000	H'FFFE4006	16
Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FFFE4008	16, 32
Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16
Timer counter U_5	TCNTU_5	R/W	H'0000	H'FFFE4080	16, 32
Timer general register U_5	TGRU_5	R/W	H'FFFF	H'FFFE4082	16
Timer control register U_5	TCRU_5	R/W	H'00	H'FFFE4084	8
Timer I/O control register U_5	TIORU_5	R/W	H'00	H'FFFE4086	8
Timer counter V_5	TCNTV_5	R/W	H'0000	H'FFFE4090	16, 32
Timer general register V_5	TGRV_5	R/W	H'FFFF	H'FFFE4092	16
Timer control register V_5	TCRV_5	R/W	H'00	H'FFFE4094	8
Timer I/O control register V_5	TIORV_5	R/W	H'00	H'FFFE4096	8
Timer counter W_5	TCNTW_5	R/W	H'0000	H'FFFE40A0	16, 32
Timer general register W_5	TGRW_5	R/W	H'FFFF	H'FFFE40A2	16
Timer control register W_5	TCRW_5	R/W	H'00	H'FFFE40A4	8
Timer I/O control register W_5	TIORW_5	R/W	H'00	H'FFFE40A6	8
Timer status register_5	TSR_5	R/W	H'00	H'FFFE40B0	8
Timer interrupt enable register_5	TIER_5	R/W	H'00	H'FFFE40B2	8
Timer start register_5	TSTR_5	R/W	H'00	H'FFFE40B4	8
Timer compare match clear register	TCNTCMPCLR	R/W	H'00	H'FFFE40B6	8

11.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The MTU2 has a total of eight TCR registers, one each for channels 0 to 4 and three (TCRU_5, TCRV_5, and TCRW_5) for channel 5. TCR register settings should be conducted only when TCNT operation is stopped.



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See tables 11.4 and 11.5 for details.
4, 3	CKEG[1:0]	00	R/W	Clock Edge 0 and 1 These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. MPϕ/4 both edges = MPϕ/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is MPϕ/4 or slower. When MPϕ/1 or the overflow/underflow of another channel is selected for the input clock, although values can be written, counter operation compiles with the initial value. 00: Count at rising edge 01: Count at falling edge 1x: Count at both edges
2 to 0	TPSC[2:0]	000	R/W	Time Prescaler 0 to 2 These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 11.6 to 11.10 for details.

[Legend]

x: Don't care

Table 11.4 CCLR0 to CCLR2 (Channels 0, 3, and 4)

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3, 4	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation is set by setting the SYNC bit in TSYR to 1.
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 11.5 CCLR0 to CCLR2 (Channels 1 and 2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: 1. Synchronous operation is selected by setting the SYNC bit in TSYR to 1.
2. Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

Table 11.6 TPSC0 to TPSC2 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 11.7 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 11.8 TPSC0 to TPSC2 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 11.9 TPSC0 to TPSC2 (Channels 3 and 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3, 4	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	Internal clock: counts on Pϕ/256
			1	Internal clock: counts on Pϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

Table 11.10 TPSC1 and TPSC0 (Channel 5)

Channel	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	Internal clock: counts on Pϕ/1
		1	Internal clock: counts on Pϕ/4
	1	0	Internal clock: counts on Pϕ/16
		1	Internal clock: counts on Pϕ/64

Note: Bits 7 to 2 are reserved in channel 5. These bits are always read as 0. The write value should always be 0.

11.3.2 Timer Mode Register (TMDR)

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode of each channel. The MTU2 has five TMDR registers, one each for channels 0 to 4. TMDR register settings should be changed only when TCNT operation is stopped.

Bit:	7	6	5	4	3	2	1	0
	-	BFE	BFB	BFA	MD[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	BFE	0	R/W	Buffer Operation E Specifies whether TGRE_0 and TGRF_0 are to operate in the normal way or to be used together for buffer operation. TGRF compare match is generated when TGRF is used as the buffer register. In channels 1 to 4, this bit is reserved. It is always read as 0 and the write value should always be 0. 0: TGRE_0 and TGRF_0 operate normally 1: TGRE_0 and TGRF_0 used together for buffer operation

Bit	Bit Name	Initial Value	R/W	Description
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated in a mode other than complementary PWM. TGRD compare match is generated in complementary PWM mode. When compare match occurs during the Tb period in complementary PWM mode, TGFD is set. Therefore, set the TGIED bit in the timer interrupt enable register 3/4 (TIER_3/4) to 0. In channels 1 and 2, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated in a mode other than complementary PWM. TGRC compare match is generated when in complementary PWM mode. When compare match for channel 4 occurs during the Tb period in complementary PWM mode, TGFC is set. Therefore, set the TGIEC bit in the timer interrupt enable register 4 (TIER_4) to 0. In channels 1 and 2, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation
3 to 0	MD[3:0]	0000	R/W	Modes 0 to 3 These bits are used to set the timer operating mode. See table 11.11 for details.

Table 11.11 Setting of Operation Mode by Bits MD0 to MD3

Bit 3 MD3	Bit 2 MD2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Setting prohibited
		1	0	PWM mode 1
			1	PWM mode 2*[1]
	1	0	0	Phase counting mode 1*[2]
			1	Phase counting mode 2*[2]
		1	0	Phase counting mode 3*[2]
			1	Phase counting mode 4*[2]
1	0	0	0	Reset synchronous PWM mode*[3]
			1	Setting prohibited
		1	X	Setting prohibited
	1	0	0	Setting prohibited
			1	Complementary PWM mode 1 (transmit at crest)*[3]
		1	0	Complementary PWM mode 2 (transmit at trough)*[3]
			1	Complementary PWM mode 2 (transmit at crest and trough)*[3]

[Legend]

X: Don't care

Notes: 1. PWM mode 2 cannot be set for channels 3 and 4.

2. Phase counting mode cannot be set for channels 0, 3, and 4.

3. Reset synchronous PWM mode, complementary PWM mode can only be set for channel 3. When channel 3 is set to reset synchronous PWM mode or complementary PWM mode, the channel 4 settings become ineffective and automatically conform to the channel 3 settings. However, do not set channel 4 to reset synchronous PWM mode or complementary PWM mode. Reset synchronous PWM mode and complementary PWM mode cannot be set for channels 0, 1, and 2.

11.3.3 Timer I/O Control Register (TIOR)

The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU2 has a total of eleven TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2, and three (TIORU_5, TIORV_5, and TIORW_5) for channel 5.

TIOR should be set while TMDR is set in 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.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4

Bit:	7	6	5	4	3	2	1	0
	IOB[3:0]				IOA[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 11.12 TIOR_1: Table 11.14 TIOR_2: Table 11.15 TIORH_3: Table 11.16 TIORH_4: Table 11.18
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify the function of TGRA. See the following tables. TIORH_0: Table 11.20 TIOR_1: Table 11.22 TIOR_2: Table 11.23 TIORH_3: Table 11.24 TIORH_4: Table 11.26

- TIORL_0, TIORL_3, TIORL_4

Bit:	7	6	5	4	3	2	1	0
	IOD[3:0]				IOC[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOD[3:0]	0000	R/W	I/O Control D0 to D3 Specify the function of TGRD. See the following tables. TIORL_0: Table 11.13 TIORL_3: Table 11.17 TIORL_4: Table 11.19
3 to 0	IOC[3:0]	0000	R/W	I/O Control C0 to C3 Specify the function of TGRC. See the following tables. TIORL_0: Table 11.21 TIORL_3: Table 11.25 TIORL_4: Table 11.27

- TIORU_5, TIORV_5, TIORW_5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	IOC[4:0]				
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	IOC[4:0]	00000	R/W	I/O Control C0 to C4 Specify the function of TGRU_5, TGRV_5, and TGRW_5. For details, see table 11.28.

Table 11.12 TIORH_0 (Channel 0)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_0 Function**	**TIOC0B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.13 TIORL_0 (Channel 0)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_0 Function**	**TIOC0D Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_0 is set to 1 and TGRD_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.14 TIOR_1 (Channel 1)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_1 Function**	**TIOC1B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of TGRC_0 compare match/input capture

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.15 TIOR_2 (Channel 2)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOC2B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.16 TIORH_3 (Channel 3)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_3 Function**	**TIOC3B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.17 TIORL_3 (Channel 3)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_3 Function**	**TIOC3D Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_3 is set to 1 and TGRD_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.18 TIORH_4 (Channel 4)

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_4 Function**	**TIOC4B Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.19 TIORL_4 (Channel 4)

				Description	
Bit 7 IOD3	**Bit 6 IOD2**	**Bit 5 IOD1**	**Bit 4 IOD0**	**TGRD_4 Function**	**TIOC4D Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFB bit in TMDR_4 is set to 1 and TGRD_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.20 TIORH_0 (Channel 0)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_0 Function**	**TIOC0A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.21 TIORL_0 (Channel 0)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_0 Function**	**TIOC0C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_0 is set to 1 and TGRC_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.22 TIOR_1 (Channel 1)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_1 Function**	**TIOC1A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	0	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges
	1	X	X		Input capture at generation of channel 0/TGRA_0 compare match/input capture

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.23 TIOR_2 (Channel 2)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_2 Function**	**TIOC2A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.24 TIORH_3 (Channel 3)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_3 Function**	**TIOC3A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.25 TIORL_3 (Channel 3)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_3 Function**	**TIOC3C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_3 is set to 1 and TGRC_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.26 TIORH_4 (Channel 4)

				Description	
Bit 3 IOA3	**Bit 2 IOA2**	**Bit 1 IOA1**	**Bit 0 IOA0**	**TGRA_4 Function**	**TIOC4A Pin Function**
0	0	0	0	Output compare register	Output retained*
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Note: * After power-on reset, 0 is output until TIOR is set.

Table 11.27 TIORL_4 (Channel 4)

				Description	
Bit 3 IOC3	**Bit 2 IOC2**	**Bit 1 IOC1**	**Bit 0 IOC0**	**TGRC_4 Function**	**TIOC4C Pin Function**
0	0	0	0	Output compare register*2	Output retained*1
			1		Initial output is 0 0 output at compare match
		1	0		Initial output is 0 1 output at compare match
			1		Initial output is 0 Toggle output at compare match
	1	0	0		Output retained
			1		Initial output is 1 0 output at compare match
		1	0		Initial output is 1 1 output at compare match
			1		Initial output is 1 Toggle output at compare match
1	X	0	0	Input capture register*2	Input capture at rising edge
			1		Input capture at falling edge
		1	X		Input capture at both edges

[Legend]

X: Don't care

Notes: 1. After power-on reset, 0 is output until TIOR is set.

2. When the BFA bit in TMDR_4 is set to 1 and TGRC_4 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 11.28 TIORU_5, TIORV_5, and TIORW_5 (Channel 5)

					Description	
Bit 4 IOC4	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRU_5, TGRV_5, and TGRW_5 Function	TIC5U, TIC5V, and TIC5W Pin Function
0	0	0	0	0	Compare match register	Compare match
				1		Setting prohibited
			1	X		Setting prohibited
		1	X	X		Setting prohibited
	1	X	X	X		Setting prohibited
1	0	0	0	0	Input capture register	Setting prohibited
				1		Input capture at rising edge
			1	0		Input capture at falling edge
				1		Input capture at both edges
		1	X	X		Setting prohibited
	1	0	0	0		Setting prohibited
				1		Measurement of low pulse width of external input signal
						Capture at trough in complementary PWM mode
			1	0		Measurement of low pulse width of external input signal
						Capture at crest in complementary PWM mode
				1		Measurement of low pulse width of external input signal
						Capture at crest and trough in complementary PWM mode
		1	0	0		Setting prohibited
				1		Measurement of high pulse width of external input signal
						Capture at trough in complementary PWM mode
			1	0		Measurement of high pulse width of external input signal
						Capture at crest in complementary PWM mode
				1		Measurement of high pulse width of external input signal
						Capture at crest and trough in complementary PWM mode

[Legend]

X: Don't care

11.3.4 Timer Compare Match Clear Register (TCNTCMPCLR)

TCNTCMPCLR is an 8-bit readable/writable register that specifies requests to clear TCNTU_5, TCNTV_5, and TCNTW_5. The MTU2 has one TCNTCMPCLR in channel 5.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMPCLR5U	CMPCLR5V	CMPCLR5W
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CMPCLR5U	0	R/W	TCNT Compare Clear 5U Enables or disables requests to clear TCNTU_5 at TGRU_5 compare match or input capture. 0: Disables TCNTU_5 to be cleared to H'0000 at TCNTU_5 and TGRU_5 compare match or input capture 1: Enables TCNTU_5 to be cleared to H'0000 at TCNTU_5 and TGRU_5 compare match or input capture
1	CMPCLR5V	0	R/W	TCNT Compare Clear 5V Enables or disables requests to clear TCNTV_5 at TGRV_5 compare match or input capture. 0: Disables TCNTV_5 to be cleared to H'0000 at TCNTV_5 and TGRV_5 compare match or input capture 1: Enables TCNTV_5 to be cleared to H'0000 at TCNTV_5 and TGRV_5 compare match or input capture

Bit	Bit Name	Initial Value	R/W	Description
0	CMPCLR5W	0	R/W	TCNT Compare Clear 5W Enables or disables requests to clear TCNTW_5 at TGRW_5 compare match or input capture. 0: Disables TCNTW_5 to be cleared to H'0000 at TCNTW_5 and TGRW_5 compare match or input capture 1: Enables TCNTW_5 to be cleared to H'0000 at TCNTW_5 and TGRW_5 compare match or input capture

11.3.5 Timer Interrupt Enable Register (TIER)

The TIER registers are 8-bit readable/writable registers that control enabling or disabling of interrupt requests for each channel. The MTU2 has seven TIER registers, two for channel 0 and one each for channels 1 to 5.

- TIER_0, TIER_1, TIER_2, TIER_3, TIER_4

Bit:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE	0	R/W	A/D Converter Start Request Enable Enables or disables generation of A/D converter start requests by TGRA input capture/compare match. 0: A/D converter start request generation disabled 1: A/D converter start request generation enabled

Bit	Bit Name	Initial Value	R/W	Description
6	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by TCNT_4 underflow (trough) in complementary PWM mode. In channels 0 to 3, bit 6 is reserved. It is always read as 0 and the write value should always be 0. 0: A/D converter start request generation by TCNT_4 underflow (trough) disabled 1: A/D converter start request generation by TCNT_4 underflow (trough) enabled
5	TCIEU	0	R/W	Underflow Interrupt Enable Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled

Bit	Bit Name	Initial Value	R/W	Description
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled

- TIER2_0

Bit:	7	6	5	4	3	2	1	0
	TTGE2	-	-	-	-	-	TGIEF	TGIEE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TTGE2	0	R/W	A/D Converter Start Request Enable 2 Enables or disables generation of A/D converter start requests by compare match between TCNT_0 and TGRE_0. 0: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 disabled 1: A/D converter start request generation by compare match between TCNT_0 and TGRE_0 enabled
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGIEF	0	R/W	TGR Interrupt Enable F Enables or disables interrupt requests by compare match between TCNT_0 and TGRF_0. 0: Interrupt requests (TGIF) by TGFE bit disabled 1: Interrupt requests (TGIF) by TGFE bit enabled
0	TGIEE	0	R/W	TGR Interrupt Enable E Enables or disables interrupt requests by compare match between TCNT_0 and TGRE_0. 0: Interrupt requests (TGIE) by TGEE bit disabled 1: Interrupt requests (TGIE) by TGEE bit enabled

- TIER_5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TGIE5U	TGIE5V	TGIE5W
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TGIE5U	0	R/W	TGR Interrupt Enable 5U Enables or disables interrupt requests (TGIU_5) by the CMFU5 bit when the CMFU5 bit in TSR_5 is set to 1. 0: Interrupt requests (TGIU_5) disabled 1: Interrupt requests (TGIU_5) enabled
1	TGIE5V	0	R/W	TGR Interrupt Enable 5V Enables or disables interrupt requests (TGIV_5) by the CMFV5 bit when the CMFV5 bit in TSR_5 is set to 1. 0: Interrupt requests (TGIV_5) disabled 1: Interrupt requests (TGIV_5) enabled
0	TGIE5W	0	R/W	TGR Interrupt Enable 5W Enables or disables interrupt requests (TGIW_5) by the CMFW5 bit when the CMFW5 bit in TSR_5 is set to 1. 0: Interrupt requests (TGIW_5) disabled 1: Interrupt requests (TGIW_5) enabled

11.3.6 Timer Status Register (TSR)

The TSR registers are 8-bit readable/writable registers that indicate the status of each channel. The MTU2 has seven TSR registers, two for channel 0 and one each for channels 1 to 5.

- TSR_0, TSR_1, TSR_2, TSR_3, TSR_4

Bit:	7	6	5	4	3	2	1	0
	TCFD	-	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7	TCFD	1	R	Count Direction Flag Status flag that shows the direction in which TCNT counts in channels 1 to 4. In channel 0, bit 7 is reserved. It is always read as 1 and the write value should always be 1. 0: TCNT counts down 1: TCNT counts up
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	TCFU	0	R/(W)*1	Underflow Flag Status flag that indicates that TCNT underflow has occurred when channels 1 and 2 are set to phase counting mode. Only 0 can be written, for flag clearing. In channels 0, 3, and 4, bit 5 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1*2 [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF)

Bit	Bit Name	Initial Value	R/W	Description
4	TCFV	0	R/(W)*1	Overflow Flag Status flag that indicates that TCNT overflow has occurred. Only 0 can be written, for flag clearing. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1*2 [Setting condition] • When the TCNT value overflows (changes from H'FFFF to H'0000) In channel 4, when the TCNT_4 value underflows (changes from H'0001 to H'0000) in complementary PWM mode, this flag is also set.
3	TGFD	0	R/(W)*1	Input Capture/Output Compare Flag D Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1*2 • When DTC is activated by TGID interrupt, and the DISEL bit of MRB in DTC is cleared to 0. [Setting conditions] • When TCNT = TGRD and TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal and TGRD is functioning as input capture register

Bit	Bit Name	Initial Value	R/W	Description
2	TGFC	0	R/(W)*1	Input Capture/Output Compare Flag C Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0, 3, and 4. Only 0 can be written, for flag clearing. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. [Clearing condition] • When DTC is activated by TGIC interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to TGFC after reading TGFC = 1*2 [Setting conditions] • When TCNT = TGRC and TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal and TGRC is functioning as input capture register
1	TGFB	0	R/(W)*1	Input Capture/Output Compare Flag B Status flag that indicates the occurrence of TGRB input capture or compare match. Only 0 can be written, for flag clearing. [Clearing condition] • When DTC is activated by TGIB interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to TGFB after reading TGFB = 1*2 [Setting conditions] • When TCNT = TGRB and TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal and TGRB is functioning as input capture register

Bit	Bit Name	Initial Value	R/W	Description
0	TGFA	0	R/(W)*1	Input Capture/Output Compare Flag A Status flag that indicates the occurrence of TGRA input capture or compare match. Only 0 can be written, for flag clearing. [Clearing conditions] • When DMAC is activated by TGIA interrupt. • When DTC is activated by TGIA interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to TGFA after reading TGFA = 1*2 [Setting conditions] • When TCNT = TGRA and TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal and TGRA is functioning as input capture register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. After reading 1, when the next flag set is generated before writing 0, the flag will not be cleared by writing 0. Read 1 again and write 0 in this case.

- TSR2_0

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TGFF	TGFE
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	TGFF	0	R/(W)*1	Compare Match Flag F Status flag that indicates the occurrence of compare match between TCNT_0 and TGRF_0. [Clearing condition] • When 0 is written to TGFF after reading TGFF = 1*2 [Setting condition] • When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register
0	TGFE	0	R/(W)*1	Compare Match Flag E Status flag that indicates the occurrence of compare match between TCNT_0 and TGRE_0. [Clearing condition] • When 0 is written to TGFE after reading TGFE = 1*2 [Setting condition] • When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. After reading 1 when the next flag set is generated before writing 0, the flag will not be cleared by writing 0. Read 1 again and write 0 in this case.

- TSR_5

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMFU5	CMFV5	CMFW5
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*1	R/(W)*1	R/(W)*1

Note: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CMFU5	0	R/(W)*1	Compare Match/Input Capture Flag U5 Status flag that indicates the occurrence of TGRU_5 input capture or compare match. [Clearing condition] • When DTC is activated by TGIU_5 interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to CMFU5 after reading CMFU5 = 1 [Setting conditions] • When TCNTU_5 = TGRU_5 and TGRU_5 is functioning as output compare register • When TCNTU_5 value is transferred to TGRU_5 by input capture signal and TGRU_5 is functioning as input capture register • When TCNTU_5 value is transferred to TGRU_5 and TGRU_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control registers U_5, V_5, and W_5 (TIORU_5, TIORV_5, and TIORW_5).*2

Bit	Bit Name	Initial Value	R/W	Description
1	CMFV5	0	R/(W)*[1]	Compare Match/Input Capture Flag V5 Status flag that indicates the occurrence of TGRV_5 input capture or compare match. [Clearing condition] • When DTC is activated by TGIV_5 interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to CMFV5 after reading CMFV5 = 1 [Setting conditions] • When TCNTV_5 = TGRV_5 and TGRV_5 is functioning as output compare register • When TCNTV_5 value is transferred to TGRV_5 by input capture signal and TGRV_5 is functioning as input capture register • When TCNTV_5 value is transferred to TGRV_5 and TGRV_5 is functioning as a register for measuring the pulse width of the external input signal. The transfer timing is specified by the IOC bits in timer I/O control registers U_5, V_5, and W_5 (TIORU_5, TIORV_5, and TIORW_5).*[2]

Bit	Bit Name	Initial Value	R/W	Description
0	CMFW5	0	R/(W)*[1]	Compare Match/Input Capture Flag W5 Status flag that indicates the occurrence of TGRW_5 input capture or compare match. Only 0 can be written to clear this flag. [Clearing condition] • When DTC is activated by TGIW_5 interrupt, and the DISEL bit of MRB in DTC is cleared to 0. • When 0 is written to CMFW5 after reading CMFW5 = 1 [Setting conditions] • When TCNTW_5 = TGRW_5 and TGRW_5 is functioning as output compare register • When TCNTW_5 value is transferred to TGRW_5 by input capture signal and TGRW_5 is functioning as input capture register • When TCNTW_5 value is transferred to TGRW_5 and TGRW_5 is functioning as a register for measuring the pulse width of the external input signal. *[2]

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Timing for transfer is set by the IOC bit in the timer I/O control register U_5/V_5/W_5 (TIORU_5/V_5/W_5).

11.3.7 Timer Buffer Operation Transfer Mode Register (TBTM)

The TBTM registers are 8-bit readable/writable registers that specify the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU2 has three TBTM registers, one each for channels 0, 3, and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TTSE	TTSB	TTSA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	TTSE	0	R/W	Timing Select E Specifies the timing for transferring data from TGRF_0 to TGRE_0 when they are used together for buffer operation. In channels 3 and 4, bit 2 is reserved. It is always read as 0 and the write value should always be 0. When channel 0 is used in a mode other than PWM mode, do not set this bit to 1. 0: When compare match E occurs in channel 0 1: When TCNT_0 is cleared
1	TTSB	0	R/W	Timing Select B Specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When the channel is used in a mode other than PWM mode, do not set this bit to 1. 0: When compare match B occurs in each channel 1: When TCNT is cleared in each channel
0	TTSA	0	R/W	Timing Select A Specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When the channel is used in a mode other than PWM mode, do not set this bit to 1. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

11.3.8 Timer Input Capture Control Register (TICCR)

TICCR is an 8-bit readable/writable register that specifies input capture conditions when TCNT_1 and TCNT_2 are cascaded. The MTU2 has one TICCR in channel 1.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	I2BE	I2AE	I1BE	I1AE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	I2BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2B pin in the TGRB_1 input capture conditions. 0: Does not include the TIOC2B pin in the TGRB_1 input capture conditions 1: Includes the TIOC2B pin in the TGRB_1 input capture conditions
2	I2AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC2A pin in the TGRA_1 input capture conditions. 0: Does not include the TIOC2A pin in the TGRA_1 input capture conditions 1: Includes the TIOC2A pin in the TGRA_1 input capture conditions
1	I1BE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1B pin in the TGRB_2 input capture conditions. 0: Does not include the TIOC1B pin in the TGRB_2 input capture conditions 1: Includes the TIOC1B pin in the TGRB_2 input capture conditions

Bit	Bit Name	Initial Value	R/W	Description
0	I1AE	0	R/W	Input Capture Enable Specifies whether to include the TIOC1A pin in the TGRA_2 input capture conditions. 0: Does not include the TIOC1A pin in the TGRA_2 input capture conditions 1: Includes the TIOC1A pin in the TGRA_2 input capture conditions

11.3.9 Timer Synchronous Clear Register S (TSYCRS)

TSYCRS is an 8-bit readable/writable register that specifies conditions for clearing TCNT_3 and TCNT_4 in the MTU2S in synchronization with the MTU2. The MTU2S has one TSYCRS in channel 3 but the MTU2 has no TSYCRS.

Bit:	7	6	5	4	3	2	1	0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CE0A	0	R/W	Clear Enable 0A Enables or disables counter clearing when the TGFA flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag in TSR_0 1: Enables counter clearing by the TGFA flag in TSR_0
6	CE0B	0	R/W	Clear Enable 0B Enables or disables counter clearing when the TGFB flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag in TSR_0 1: Enables counter clearing by the TGFB flag in TSR_0

Bit	Bit Name	Initial Value	R/W	Description
5	CE0C	0	R/W	Clear Enable 0C Enables or disables counter clearing when the TGFC flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFC flag in TSR_0 1: Enables counter clearing by the TGFC flag in TSR_0
4	CE0D	0	R/W	Clear Enable 0D Enables or disables counter clearing when the TGFD flag of TSR_0 in the MTU2 is set. 0: Disables counter clearing by the TGFD flag in TSR_0 1: Enables counter clearing by the TGFD flag in TSR_0
3	CE1A	0	R/W	Clear Enable 1A Enables or disables counter clearing when the TGFA flag of TSR_1 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag in TSR_1 1: Enables counter clearing by the TGFA flag in TSR_1
2	CE1B	0	R/W	Clear Enable 1B Enables or disables counter clearing when the TGFB flag of TSR_1 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag in TSR_1 1: Enables counter clearing by the TGFB flag in TSR_1
1	CE2A	0	R/W	Clear Enable 2A Enables or disables counter clearing when the TGFA flag of TSR_2 in the MTU2 is set. 0: Disables counter clearing by the TGFA flag in TSR_2 1: Enables counter clearing by the TGFA flag in TSR_2
0	CE2B	0	R/W	Clear Enable 2B Enables or disables counter clearing when the TGFB flag of TSR_2 in the MTU2 is set. 0: Disables counter clearing by the TGFB flag in TSR_2 1: Enables counter clearing by the TGFB flag in TSR_2

11.3.10 Timer A/D Converter Start Request Control Register (TADCR)

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 MTU2 has one TADCR in channel 4.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BF[1:0]		-	-	-	-	-	-	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Initial value:	0	0	0	0	0	0	0	0	0	0*	0	0*	0*	0*	0*	0*
R/W:	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
15, 14	BF[1:0]	00	R/W	TADCOBRA_4/TADCOBRB_4 Transfer Timing Select Select the timing for transferring data from TADCOBRA_4 and TADCOBRB_4 to TADCORA_4 and TADCORB_4. For details, see table 11.29.
13 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	UT4AE	0	R/W	Up-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 up-count operation
6	DT4AE	0*	R/W	Down-Count TRG4AN Enable Enables or disables A/D converter start requests (TRG4AN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4AN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4AN) enabled during TCNT_4 down-count operation

Bit	Bit Name	Initial Value	R/W	Description
5	UT4BE	0	R/W	Up-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 up-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 up-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 up-count operation
4	DT4BE	0*	R/W	Down-Count TRG4BN Enable Enables or disables A/D converter start requests (TRG4BN) during TCNT_4 down-count operation. 0: A/D converter start requests (TRG4BN) disabled during TCNT_4 down-count operation 1: A/D converter start requests (TRG4BN) enabled during TCNT_4 down-count operation
3	ITA3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping
2	ITA4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4AN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping
1	ITB3AE	0*	R/W	TGIA_3 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TGIA_3 interrupt skipping operation. 0: Does not link with TGIA_3 interrupt skipping 1: Links with TGIA_3 interrupt skipping

Bit	Bit Name	Initial Value	R/W	Description
0	ITB4VE	0*	R/W	TCIV_4 Interrupt Skipping Link Enable Select whether to link A/D converter start requests (TRG4BN) with TCIV_4 interrupt skipping operation. 0: Does not link with TCIV_4 interrupt skipping 1: Links with TCIV_4 interrupt skipping

Notes: 1. TADCR must not be accessed in eight bits; it should always be accessed in 16 bits.
2. 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 (3ACOR and 4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).
3. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
* Do not set to 1 when complementary PWM mode is not selected.

Table 11.29 Setting of Transfer Timing by Bits BF1 and BF0

Bit 7	Bit 6	
BF1	**BF0**	**Description**
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 TCNT_4 count.*[1]
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the TCNT_4 count.*[2]
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the TCNT_4 count.*[2]

Notes: 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the TCNT_4 count is reached in complementary PWM mode, when compare match occurs between TCNT_3 and TGRA_3 in reset-synchronized PWM mode, or when compare match occurs between TCNT_4 and TGRA_4 in PWM mode 1 or normal operation mode.
2. These settings are prohibited when complementary PWM mode is not selected.

11.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4)

TADCORA_4 and TADCORB_4 are 16-bit readable/writable registers. When the TCNT_4 count reaches the value in TADCORA_4 or TADCORB_4, a corresponding A/D converter start request will be issued.

TADCORA_4 and TADCORB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCORA_4 and TADCORB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

11.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4)

TADCOBRA_4 and TADCOBRB_4 are 16-bit readable/writable registers. When the crest or trough of the TCNT_4 count is reached, these register values are transferred to TADCORA_4 and TADCORB_4, respectively.

TADCOBRA_4 and TADCOBRB_4 are initialized to H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TADCOBRA_4 and TADCOBRB_4 must not be accessed in eight bits; they should always be accessed in 16 bits.

11.3.13 Timer Counter (TCNT)

The TCNT counters are 16-bit readable/writable counters. The MTU2 has eight TCNT counters, one each for channels 0 to 4 and three (TCNTU_5, TCNTV_5, and TCNTW_5) for channel 5.

The TCNT counters are initialized to H'0000 by a reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TCNT counters must not be accessed in eight bits; they should always be accessed in 16 bits.

11.3.14 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers. The MTU2 has 21 TGR registers, six for channel 0, two each for channels 1 and 2, four each for channels 3 and 4, and three for channel 5.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD for channels 0, 3, and 4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

TGRE_0 and TGRF_0 function as compare registers. When the TCNT_0 count matches the TGRE_0 value, an A/D converter start request can be issued. TGRF can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.

TGRU_5, TGRV_5, and TGRW_5 function as compare match, input capture, or external pulse width measurement registers.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: The TGR registers must not be accessed in eight bits; they should always be accessed in 16 bits.
TGR registers are initialized to H'FFFF.

11.3.15 Timer Start Register (TSTR)

TSTR is an 8-bit readable/writable register that selects operation/stoppage of TCNT for channels 0 to 4.

TSTR_5 is an 8-bit readable/writable register that selects operation/stoppage of TCNTU_5, TCNTV_5, and TCNTW_5 for channel 5.

When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

- TSTR

Bit:	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CST4	0	R/W	Counter Start 4 and 3
6	CST3	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_4 and TCNT_3 count operation is stopped 1: TCNT_4 and TCNT_3 performs count operation
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	CST2	0	R/W	Counter Start 2 to 0
1	CST1	0	R/W	These bits select operation or stoppage for TCNT.
0	CST0	0	R/W	If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation

- TSTR_5

Bit :	7	6	5	4	3	2	1	0
	-	-	-	-	-	CSTU5	CSTV5	CSTW5
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CSTU5	0	R/W	Counter Start U5 Selects operation or stoppage for TCNTU_5. 0: TCNTU_5 count operation is stopped 1: TCNTU_5 performs count operation
1	CSTV5	0	R/W	Counter Start V5 Selects operation or stoppage for TCNTV_5. 0: TCNTV_5 count operation is stopped 1: TCNTV_5 performs count operation
0	CSTW5	0	R/W	Counter Start W5 Selects operation or stoppage for TCNTW_5. 0: TCNTW_5 count operation is stopped 1: TCNTW_5 performs count operation

11.3.16 Timer Synchronous Register (TSYR)

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	-	-	-	SYNC2	SYNC1	SYNC0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SYNC4	0	R/W	Timer Synchronous operation 4 and 3
6	SYNC3	0	R/W	These bits are used to select 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 CCLR0 to CCLR2 in TCR. 0: TCNT_4 and TCNT_3 operate independently (TCNT presetting/clearing is unrelated to other channels) 1: TCNT_4 and TCNT_3 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible
5 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	SYNC2	0	R/W	Timer Synchronous operation 2 to 0
1	SYNC1	0	R/W	These bits are used to select whether operation is independent of or synchronized with other channels.
0	SYNC0	0	R/W	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 CCLR0 to CCLR2 in TCR. 0: TCNT_2 to TCNT_0 operates independently (TCNT presetting /clearing is unrelated to other channels) 1: TCNT_2 to TCNT_0 performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

11.3.17 Timer Counter Synchronous Start Register (TCSYSTR)

TCSYSTR is an 8-bit readable/writable register that specifies synchronous start of the MTU2 and MTU2S counters. Note that the MTU2S does not have TCSYSTR.

Bit:	7	6	5	4	3	2	1	0
	SCH0	SCH1	SCH2	SCH3	SCH4	-	SCH3S	SCH4S
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*

Note: * Only 1 can be written to set the register.

Bit	Bit Name	Initial Value	R/W	Description
7	SCH0	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_0 in the MTU2. 0: Does not specify synchronous start for TCNT_0 in the MTU2 1: Specifies synchronous start for TCNT_0 in the MTU2 [Clearing condition] • When 1 is set to the CST0 bit of TSTR in MTU2 while SCH0 = 1
6	SCH1	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_1 in the MTU2. 0: Does not specify synchronous start for TCNT_1 in the MTU2 1: Specifies synchronous start for TCNT_1 in the MTU2 [Clearing condition] • When 1 is set to the CST1 bit of TSTR in MTU2 while SCH1 = 1

Bit	Bit Name	Initial Value	R/W	Description
5	SCH2	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_2 in the MTU2. 0: Does not specify synchronous start for TCNT_2 in the MTU2 1: Specifies synchronous start for TCNT_2 in the MTU2 [Clearing condition] • When 1 is set to the CST2 bit of TSTR in MTU2 while SCH2 = 1
4	SCH3	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_3 in the MTU2. 0: Does not specify synchronous start for TCNT_3 in the MTU2 1: Specifies synchronous start for TCNT_3 in the MTU2 [Clearing condition] • When 1 is set to the CST3 bit of TSTR in MTU2 while SCH3 = 1
3	SCH4	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4 in the MTU2. 0: Does not specify synchronous start for TCNT_4 in the MTU2 1: Specifies synchronous start for TCNT_4 in the MTU2 [Clearing condition] • When 1 is set to the CST4 bit of TSTR in MTU2 while SCH4 = 1
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	SCH3S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_3S in the MTU2S. 0: Does not specify synchronous start for TCNT_3S in the MTU2S 1: Specifies synchronous start for TCNT_3S in the MTU2S [Clearing condition] • When 1 is set to the CST3 bit of TSTRS in MTU2S while SCH3S = 1
0	SCH4S	0	R/(W)*	Synchronous Start Controls synchronous start of TCNT_4S in the MTU2S. 0: Does not specify synchronous start for TCNT_4S in the MTU2S 1: Specifies synchronous start for TCNT_4S in the MTU2S [Clearing condition] • When 1 is set to the CST4 bit of TSTRS in MTU2S while SCH4S = 1

Note: Only 1 can be written to set the register.

11.3.18 Timer Read/Write Enable Register (TRWER)

TRWER is an 8-bit readable/writable register that enables or disables access to the registers and counters which have write-protection capability against accidental modification in channels 3 and 4.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RWE
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RWE	1	R/W	Read/Write Enable Enables or disables access to the registers which have write-protection capability against accidental modification. 0: Disables read/write access to the registers 1: Enables read/write access to the registers [Clearing condition] • When 0 is written to the RWE bit after reading RWE = 1

- Registers and counters having write-protection capability against accidental modification
  22 registers: TCR_3, TCR_4, TMDR_3, TMDR_4, TIORH_3, TIORH_4, TIORL_3, TIORL_4, TIER_3, TIER_4, TGRA_3, TGRA_4, TGRB_3, TGRB_4, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, TCNT_3, and TCNT4.

11.3.19 Timer Output Master Enable Register (TOER)

TOER is an 8-bit readable/writable register that enables/disables output settings for output pins TIOC4D, TIOC4C, TIOC3D, TIOC4B, TIOC4A, and TIOC3B. These pins do not output correctly if the TOER bits have not been set. Set TOER of CH3 and CH4 prior to setting TIOR of CH3 and CH4.

Bit:	7	6	5	4	3	2	1	0
	-	-	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Initial value:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	OE4D	0	R/W	Master Enable TIOC4D This bit enables/disables the TIOC4D pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
4	OE4C	0	R/W	Master Enable TIOC4C This bit enables/disables the TIOC4C pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
3	OE3D	0	R/W	Master Enable TIOC3D This bit enables/disables the TIOC3D pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
2	OE4B	0	R/W	Master Enable TIOC4B This bit enables/disables the TIOC4B pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled
1	OE4A	0	R/W	Master Enable TIOC4A This bit enables/disables the TIOC4A pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

Bit	Bit Name	Initial Value	R/W	Description
0	OE3B	0	R/W	Master Enable TIOC3B This bit enables/disables the TIOC3B pin MTU2 output. 0: MTU2 output is disabled (inactive level)* 1: MTU2 output is enabled

Note: * The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 11.3.20, Timer Output Control Register 1 (TOCR1), and section 11.3.21, Timer Output Control Register 2 (TOCR2). Set these bits to 1 to enable MTU2 output in other than complementary PWM or reset-synchronized PWM mode. When these bits are set to 0, low level is output.

11.3.20 Timer Output Control Register 1 (TOCR1)

TOCR1 is an 8-bit readable/writable register that enables/disables PWM synchronized toggle output in complementary PWM mode/reset synchronized PWM mode, and controls output level inversion of PWM output.

Bit:	7	6	5	4	3	2	1	0
	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*	R/W	R/W	R/W

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	Bit Name	Initial value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	PSYE	0	R/W	PWM Synchronous Output Enable This bit selects the enable/disable of toggle output synchronized with the PWM period. 0: Toggle output is disabled 1: Toggle output is enabled
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial value	R/W	Description
3	TOCL	0	R/(W)*	TOC Register Write Protection*[1] This bit selects the enable/disable of write access to the TOCS, OLSN, and OLSP bits in TOCR1. 0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled
2	TOCS	0	R/W	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. 0: TOCR1 setting is selected 1: TOCR2 setting is selected
1	OLSN	0	R/W	Output Level Select N*[2]*[3] This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.30.
0	OLSP	0	R/W	Output Level Select P*[2]*[3] This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 11.31.

Notes: 1. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
2. Clearing the TOCS0 bit to 0 makes this bit setting valid.
3. The inverse-phase output is the exact inverse of the positive-phase output unless dead time is generated. When no dead time is generated, only the OLSP setting is valid.

Table 11.30 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to active level after elapse of the dead time after count start.

Table 11.31 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Figure 11.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

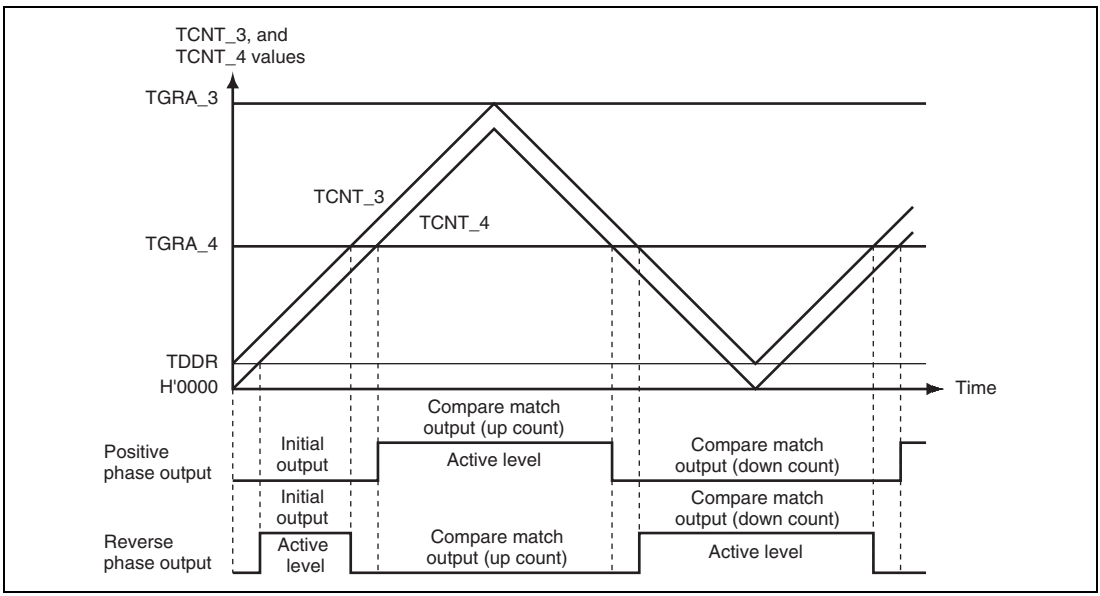


Figure 11.2 Complementary PWM Mode Output Level Example

11.3.21 Timer Output Control Register 2 (TOCR2)

TOCR2 is an 8-bit readable/writable register that controls output level inversion of PWM output in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	BF[1:0]	00	R/W	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBR to TOCR2. For details, see table 11.32.
5	OLS3N	0	R/W	Output Level Select 3N*1*2 This bit selects the output level on TIOC4D in reset-synchronized PWM mode/complementary PWM mode. See table 11.33.
4	OLS3P	0	R/W	Output Level Select 3P*1*2 This bit selects the output level on TIOC4B in reset-synchronized PWM mode/complementary PWM mode. See table 11.34.
3	OLS2N	0	R/W	Output Level Select 2N*1*2 This bit selects the output level on TIOC4C in reset-synchronized PWM mode/complementary PWM mode. See table 11.35.
2	OLS2P	0	R/W	Output Level Select 2P*1*2 This bit selects the output level on TIOC4A in reset-synchronized PWM mode/complementary PWM mode. See table 11.36.
1	OLS1N	0	R/W	Output Level Select 1N*1*2 This bit selects the output level on TIOC3D in reset-synchronized PWM mode/complementary PWM mode. See table 11.37.

Bit	Bit Name	Initial value	R/W	Description
0	OLS1P	0	R/W	Output Level Select 1P*1*2 This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 11.38.

Notes: 1. Setting the TOCS bit in TOCR1 to 1 makes this bit setting valid.
2. The inverse-phase output is the exact inverse of the positive-phase output unless dead time is generated. When no dead time is generated, only the OLSiP setting is valid.

Table 11.32 Setting of Bits BF1 and BF0

Bit 7	Bit 6	Description	
BF1	BF0	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the TCNT_4 count.	Transfers data from the buffer register (TOLBR) to TOCR2 when TCNT_3/TCNT_4 is cleared
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the TCNT_4 count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the TCNT_4 count.	Setting prohibited

Table 11.33 TIOC4D Output Level Select Function

Bit 5	Function			
			Compare Match Output	
OLS3N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.34 TIOC4B Output Level Select Function

Bit 4	Function			
			Compare Match Output	
OLS3P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 11.35 TIOC4C Output Level Select Function

Bit 3	Function			
			Compare Match Output	
OLS2N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.36 TIOC4A Output Level Select Function

Bit 2	Function			
			Compare Match Output	
OLS2P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 11.37 TIOC3D Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLS1N	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The reverse phase waveform initial output value changes to the active level after elapse of the dead time after count start.

Table 11.38 TIOC3B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

11.3.22 Timer Output Level Buffer Register (TOLBR)

TOLBR is an 8-bit readable/writable register that functions as a buffer for TOCR2 and specifies the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	OLS3N	0	R/W	Specifies the buffer value to be transferred to the OLS3N bit in TOCR2.
4	OLS3P	0	R/W	Specifies the buffer value to be transferred to the OLS3P bit in TOCR2.
3	OLS2N	0	R/W	Specifies the buffer value to be transferred to the OLS2N bit in TOCR2.
2	OLS2P	0	R/W	Specifies the buffer value to be transferred to the OLS2P bit in TOCR2.
1	OLS1N	0	R/W	Specifies the buffer value to be transferred to the OLS1N bit in TOCR2.
0	OLS1P	0	R/W	Specifies the buffer value to be transferred to the OLS1P bit in TOCR2.

Figure 11.3 shows an example of the PWM output level setting procedure in buffer operation.

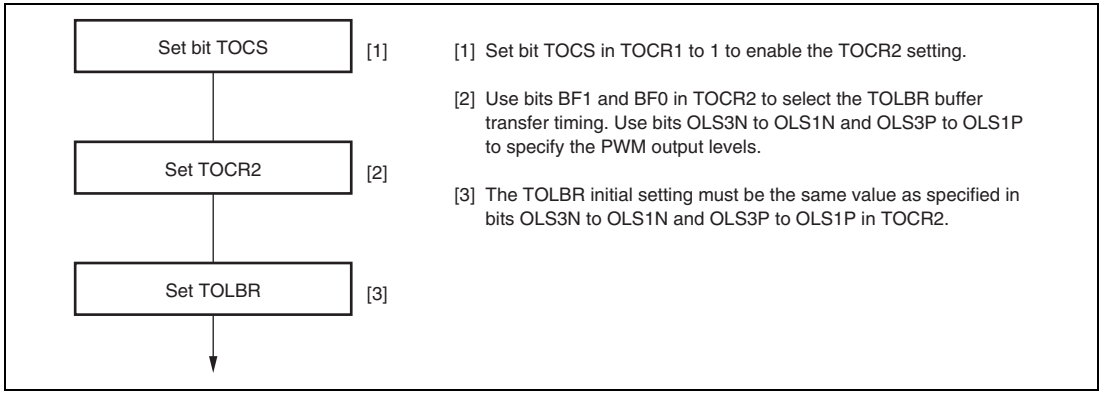


Figure 11.3 PWM Output Level Setting Procedure in Buffer Operation

11.3.23 Timer Gate Control Register (TGCR)

TGCR is an 8-bit readable/writable register that controls the waveform output necessary for brushless DC motor control in reset-synchronized PWM mode/complementary PWM mode. These register settings are ineffective for anything other than complementary PWM mode/reset-synchronized PWM mode.

Bit:	7	6	5	4	3	2	1	0
	-	BDC	N	P	FB	WF	VF	UF
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
6	BDC	0	R/W	Brushless DC Motor This bit selects whether to make the functions of this register (TGCR) effective or ineffective. 0: Ordinary output 1: Functions of this register are made effective

Bit	Bit Name	Initial value	R/W	Description
5	N	0	R/W	Reverse Phase Output (N) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the reverse pins (TIOC3D, TIOC4C, and TIOC4D) are output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
4	P	0	R/W	Positive Phase Output (P) Control This bit selects whether the level output or the reset-synchronized PWM/complementary PWM output while the positive pin (TIOC3B, TIOC4A, and TIOC4B) are output. 0: Level output 1: Reset synchronized PWM/complementary PWM output
3	FB	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the output of the positive/reverse phase is carried out automatically with the MTU2/channel 0 TGRA, TGRB, TGRC input capture signals or by writing 0 or 1 to bits 2 to 0 in TGCR. 0: Output switching is external input (Input sources are channel 0 TGRA, TGRB, TGRC input capture signal) 1: Output switching is carried out by software (setting values of UF, VF, and WF in TGCR).
2	WF	0	R/W	Output Phase Switch 2 to 0 These bits set the positive phase/negative phase output phase on or off state. The setting of these bits is valid only when the FB bit in this register is set to 1. In this case, the setting of bits 2 to 0 is a substitute for external input. See table 11.39.
1	VF	0	R/W	
0	UF	0	R/W	

Note: Do not set the FB bit to 0 when the BDC bit in MTU2S has been set to 1.

Table 11.39 Output level Select Function

			Function					
Bit 2	**Bit 1**	**Bit 0**	**TIOC3B**	**TIOC4A**	**TIOC4B**	**TIOC3D**	**TIOC4C**	**TIOC4D**
WF	**VF**	**UF**	**U Phase**	**V Phase**	**W Phase**	**U Phase**	**V Phase**	**W Phase**
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
		1	ON	OFF	OFF	OFF	OFF	ON
	1	0	OFF	ON	OFF	ON	OFF	OFF
		1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
		1	ON	OFF	OFF	OFF	ON	OFF
	1	0	OFF	OFF	ON	ON	OFF	OFF
		1	OFF	OFF	OFF	OFF	OFF	OFF

11.3.24 Timer Subcounter (TCNTS)

TCNTS is a 16-bit read-only counter that is used only in complementary PWM mode.

The initial value of TCNTS is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

11.3.25 Timer Dead Time Data Register (TDDR)

TDDR is a 16-bit register, used only in complementary PWM mode that specifies the TCNT_3 and TCNT_4 counter offset values. In complementary PWM mode, when the TCNT_3 and TCNT_4 counters are cleared and then restarted, the TDDR register value is loaded into the TCNT_3 counter and the count operation starts.

The initial value of TDDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

11.3.26 Timer Cycle Data Register (TCDR)

TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value (note that this value should be at least double the value specified in TDDR + 3) as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment).

The initial value of TCDR is H'FFFF.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

11.3.27 Timer Cycle Buffer Register (TCBR)

TCBR is a 16-bit register used only in complementary PWM mode. It functions as a buffer register for the TCDR register. The TCBR register values are transferred to the TCDR register with the transfer timing set in the TMDR register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

11.3.28 Timer Interrupt Skipping Set Register (TITCR)

TITCR is an 8-bit readable/writable register that enables or disables interrupt skipping and specifies the interrupt skipping count. The MTU2 has one TITCR.

Bit:	7	6	5	4	3	2	1	0
	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	T3AEN	0	R/W	T3AEN Enables or disables TGIA_3 interrupt skipping. 0: TGIA_3 interrupt skipping disabled 1: TGIA_3 interrupt skipping enabled
6 to 4	3ACOR[2:0]	000	R/W	These bits specify the TGIA_3 interrupt skipping count within the range from 0 to 7.* For details, see table 11.40.
3	T4VEN	0	R/W	T4VEN Enables or disables TCIV_4 interrupt skipping. 0: TCIV_4 interrupt skipping disabled 1: TCIV_4 interrupt skipping enabled

Bit	Bit Name	Initial value	R/W	Description
2 to 0	4VCOR[2:0]	000	R/W	These bits specify the TCIV_4 interrupt skipping count within the range from 0 to 7.* For details, see table 11.41.

Note: * When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter (TICNT).

Table 11.40 Setting of Interrupt Skipping Count by Bits 3ACOR2 to 3ACOR0

Bit 6	Bit 5	Bit 4	
3ACOR2	3ACOR1	3ACOR0	Description
0	0	0	Does not skip TGIA_3 interrupts.
0	0	1	Sets the TGIA_3 interrupt skipping count to 1.
0	1	0	Sets the TGIA_3 interrupt skipping count to 2.
0	1	1	Sets the TGIA_3 interrupt skipping count to 3.
1	0	0	Sets the TGIA_3 interrupt skipping count to 4.
1	0	1	Sets the TGIA_3 interrupt skipping count to 5.
1	1	0	Sets the TGIA_3 interrupt skipping count to 6.
1	1	1	Sets the TGIA_3 interrupt skipping count to 7.

Table 11.41 Setting of Interrupt Skipping Count by Bits 4VCOR2 to 4VCOR0

Bit 2	Bit 1	Bit 0	
4VCOR2	4VCOR1	4VCOR0	Description
0	0	0	Does not skip TCIV_4 interrupts.
0	0	1	Sets the TCIV_4 interrupt skipping count to 1.
0	1	0	Sets the TCIV_4 interrupt skipping count to 2.
0	1	1	Sets the TCIV_4 interrupt skipping count to 3.
1	0	0	Sets the TCIV_4 interrupt skipping count to 4.
1	0	1	Sets the TCIV_4 interrupt skipping count to 5.
1	1	0	Sets the TCIV_4 interrupt skipping count to 6.
1	1	1	Sets the TCIV_4 interrupt skipping count to 7.

11.3.29 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable/writable counter. The MTU2 has one TITCNT. TITCNT retains its value even after stopping the count operation of TCNT_3 and TCNT_4.

Bit:	7	6	5	4	3	2	1	0
	-	3ACNT[2:0]			-	4VCNT[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0.
6 to 4	3ACNT[2:0]	000	R	TGIA_3 Interrupt Counter While the T3AEN bit in TITCR is set to 1, the count in these bits is incremented every time a TGIA_3 interrupt occurs. [Clearing conditions] • When the 3ACNT2 to 3ACNT0 value in TITCNT matches the 3ACOR2 to 3ACOR0 value in TITCR • When the T3AEN bit in TITCR is cleared to 0 • When the 3ACOR2 to 3ACOR0 bits in TITCR are cleared to 0
3	—	0	R	Reserved This bit is always read as 0.
2 to 0	4VCNT[2:0]	000	R	TCIV_4 Interrupt Counter While the T4VEN bit in TITCR is set to 1, the count in these bits is incremented every time a TCIV_4 interrupt occurs. [Clearing conditions] • When the 4VCNT2 to 4VCNT0 value in TITCNT matches the 4VCOR2 to 4VCOR2 value in TITCR • When the T4VEN bit in TITCR is cleared to 0 • When the 4VCOR2 to 4VCOR2 bits in TITCR are cleared to 0

Note: To clear the TITCNT, clear the bits T3AEN and T4VEN in TITCR to 0.

11.3.30 Timer Buffer Transfer Set Register (TBTER)

TBTER is an 8-bit readable/writable register that enables or disables transfer from the buffer registers* used in complementary PWM mode to the temporary registers and specifies whether to link the transfer with interrupt skipping operation. The MTU2 has one TBTER.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	BTE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	BTE[1:0]	00	R/W	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. For details, see table 11.42.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Table 11.42 Setting of Bits BTE1 and BTE0

Bit 1	Bit 0	
BTE1	**BTE0**	**Description**
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 11.4.8, Complementary PWM Mode.

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 (3ACOR and 4VCOR) in TITCR are cleared to 0)), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

11.3.31 Timer Dead Time Enable Register (TDER)

TDER is an 8-bit readable/writable register that controls dead time generation in complementary PWM mode. The MTU2 has one TDER in channel 3. TDER must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TDER
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/(W)

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDER	1	R/(W)	Dead Time Enable Specifies whether to generate dead time. 0: Does not generate dead time 1: Generates dead time* [Clearing condition] • When 0 is written to TDER after reading TDER = 1

Note: * TDDR must be set to 1 or a larger value.

11.3.32 Timer Waveform Control Register (TWCR)

TWCR is an 8-bit readable/writable register that controls the waveform when synchronous counter clearing occurs in TCNT_3 and TCNT_4 in complementary PWM mode and specifies whether to clear the counters at TGRA_3 compare match. The CCE bit and WRE bit in TWCR must be modified only while TCNT stops.

Bit:	7	6	5	4	3	2	1	0
	CCE	-	-	-	-	-	SCC	WRE
Initial value:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R/(W)	R/(W)

Note: * Do not set to 1 when complementary PWM mode is not selected.

Bit	Bit Name	Initial Value	R/W	Description
7	CCE	0*	R/(W)	Compare Match Clear Enable Specifies whether to clear counters at TGRA_3 compare match in complementary PWM mode. 0: Does not clear counters at TGRA_3 compare match 1: Clears counters at TGRA_3 compare match [Setting condition] • When 1 is written to CCE after reading CCE = 0
6 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	SCC	0	R/(W)	Synchronous Clearing Control Specifies whether to clear TCNT_3 and TCNT_4 in the MTU2S when synchronous counter clearing between the MTU2 and MTU2S occurs in complementary PWM mode. When using this control, place the MTU2S in complementary PWM mode. When modifying the SCC bit while the counters are operating, do not modify the CCE or WRE bits. Counter clearing synchronized with the MTU2 is disabled by the SCC bit setting only when synchronous clearing occurs outside the Tb interval at the trough. When synchronous clearing occurs in the Tb interval at the trough including the period immediately after TCNT_3 and TCNT_4 start operation, TCNT_3 and TCNT_4 in the MTU2S are cleared. For the Tb interval at the trough in complementary PWM mode, see figure 11.40. In the MTU2, this bit is reserved. It is always read as 0 and the write value should always be 0. 0: Enables clearing of TCNT_3 and TCNT_4 in the MTU2S by MTU2-MTU2S synchronous clearing operation 1: Disables clearing of TCNT_3 and TCNT_4 in the MTU2S by MTU2-MTU2S synchronous clearing operation [Setting condition] • When 1 is written to SCC after reading SCC = 0

Bit	Bit Name	Initial Value	R/W	Description
0	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 11.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Note: * Do not set to 1 when complementary PWM mode is not selected.

11.3.33 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCOR), and timer A/D converter start request cycle set buffer registers (TADCOBR) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/writes. 8-bit read/write is not possible. Always access in 16-bit units.

All registers other than the above registers are 8-bit registers. These are connected to the CPU by a 16-bit data bus, so 16-bit read/writes and 8-bit read/writes are both possible.

11.4 Operation

11.4.1 Basic Functions

Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, cycle counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Always select MTU2 external pins set function using the pin function controller (PFC).

(1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in TSTR_5 is set to 1, the TCNT counter for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 11.4 shows an example of the count operation setting procedure.

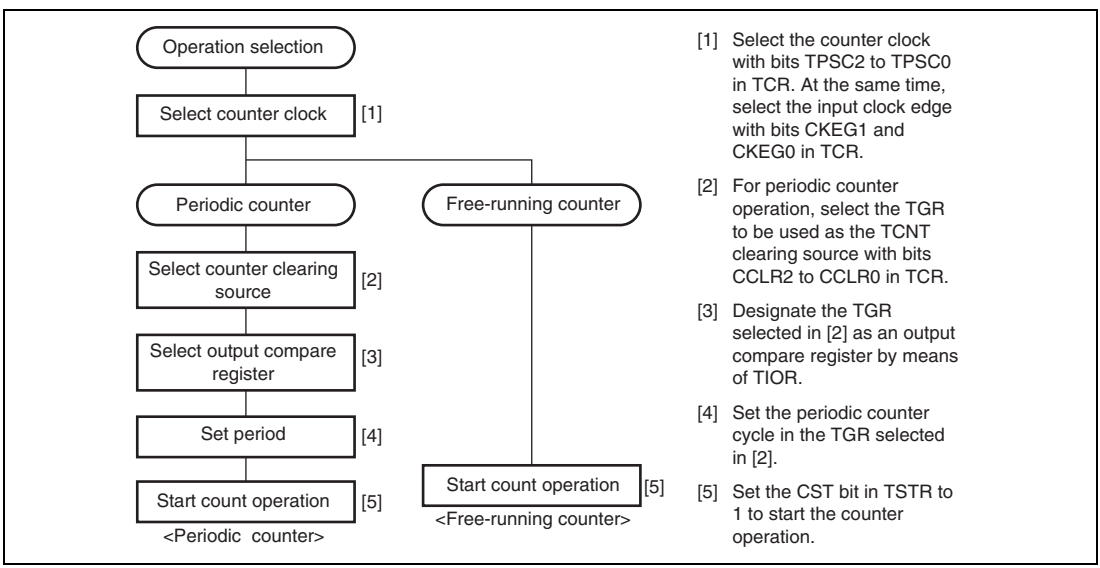


Figure 11.4 Example of Counter Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation:

Immediately after a reset, the MTU2'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 H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the MTU2 requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 11.5 illustrates free-running counter operation.

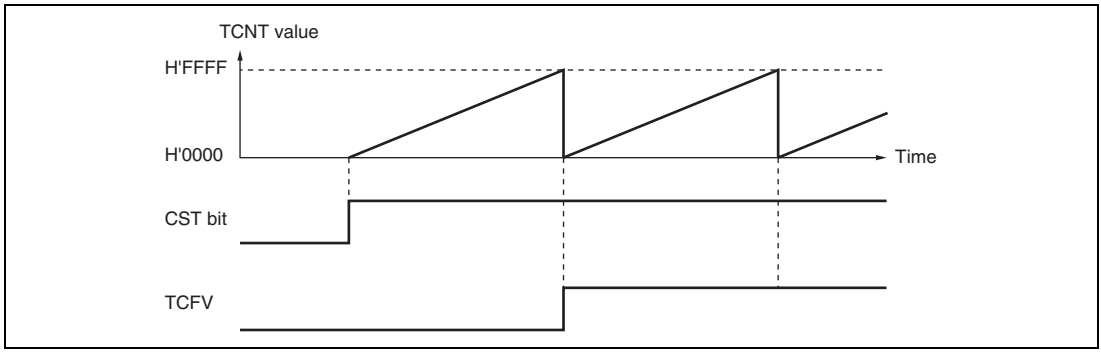


Figure 11.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR0 to CCLR2 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 value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU2 requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 11.6 illustrates periodic counter operation.

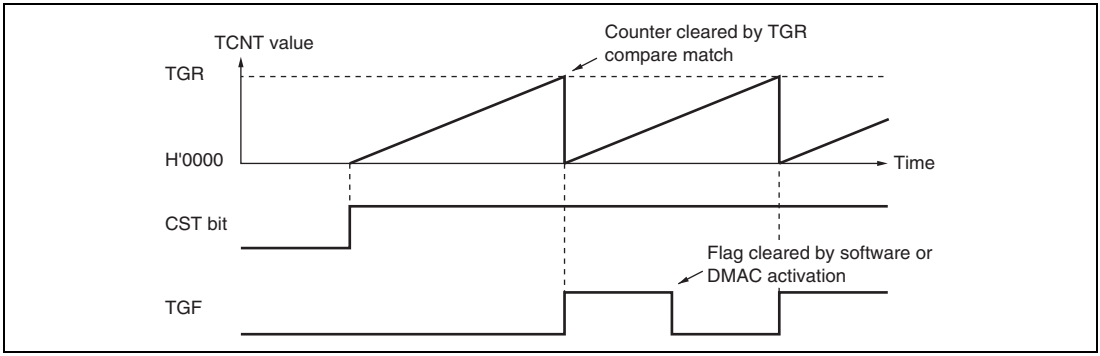


Figure 11.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU2 can perform 0, 1, or toggle output from the corresponding output pin using compare match.

(a) Example of Setting Procedure for Waveform Output by Compare Match

Figure 11.7 shows an example of the setting procedure for waveform output by compare match.

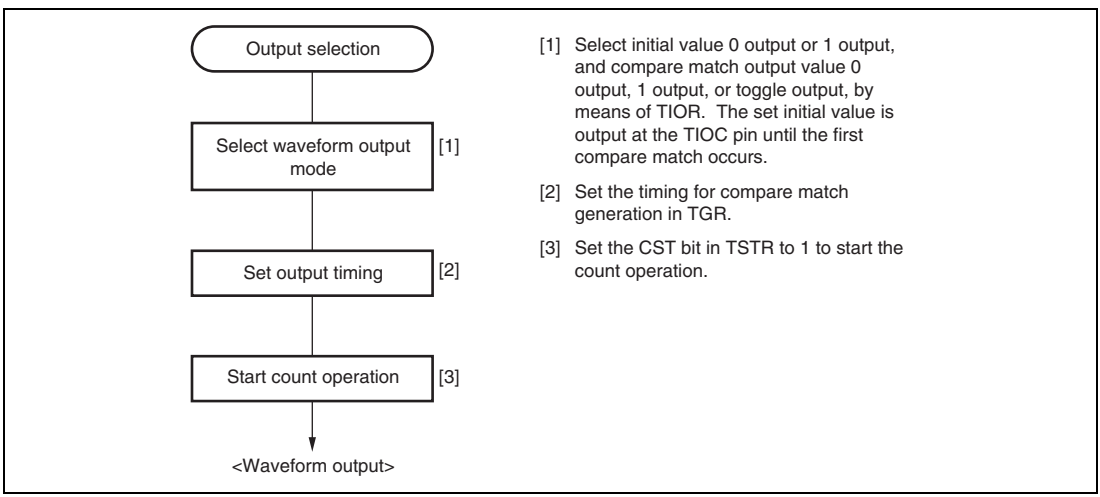


Figure 11.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 11.8 shows an example of 0 output/1 output.

In this example TCNT has been designated as a free-running counter, and settings have been made such that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

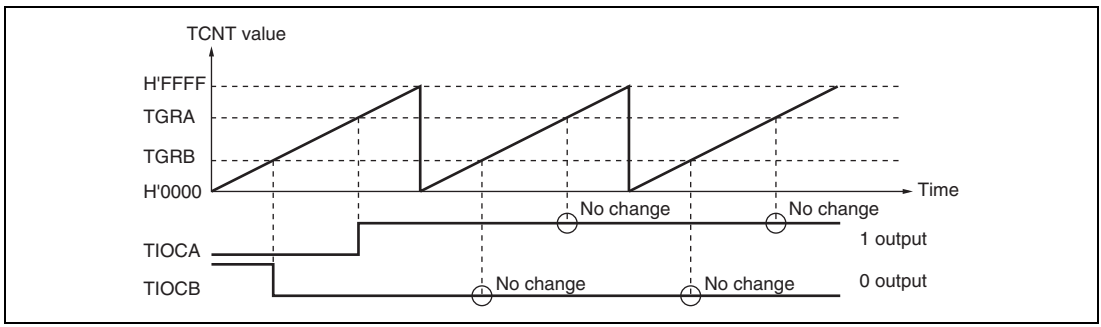


Figure 11.8 Example of 0 Output/1 Output Operation

Figure 11.9 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made such that the output is toggled by both compare match A and compare match B.

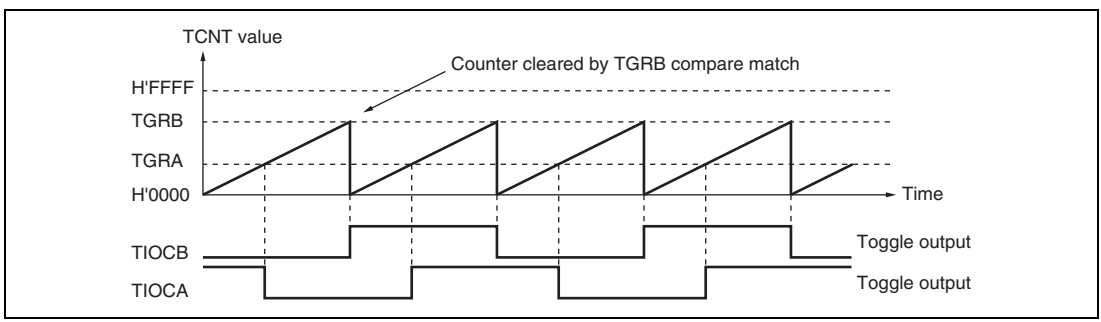


Figure 11.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, Pϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Pϕ/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 11.10 shows an example of the input capture operation setting procedure.

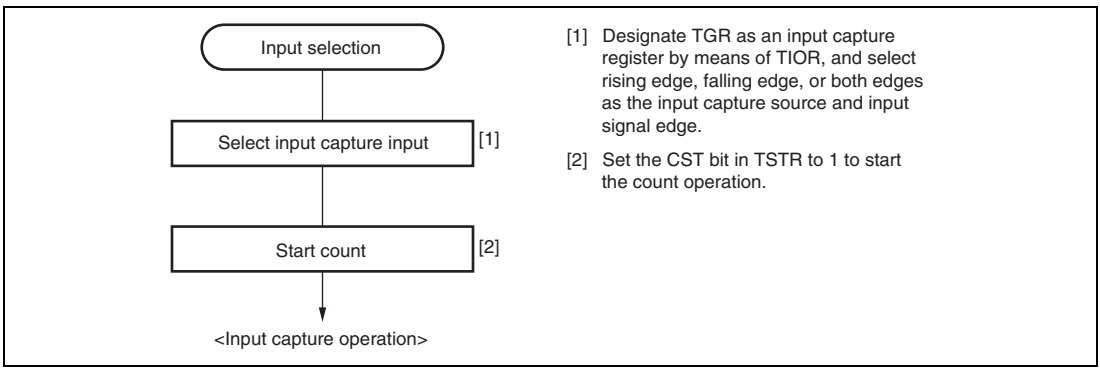


Figure 11.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 11.11 shows an example of input capture operation.

In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, the falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

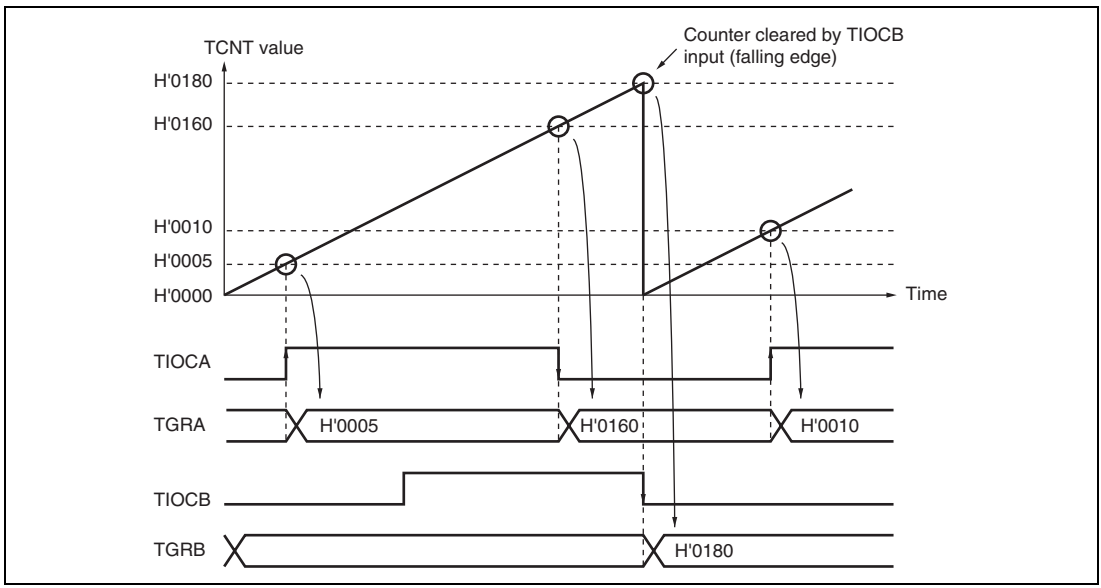


Figure 11.11 Example of Input Capture Operation

11.4.2 Synchronous Operation

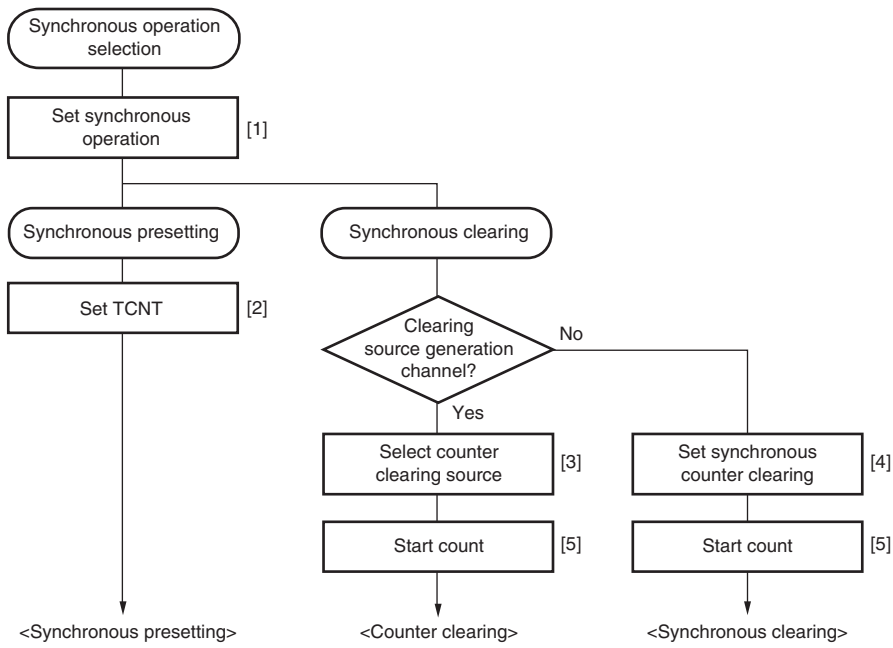
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 4 can all be designated for synchronous operation. Channel 5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 11.12 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 11.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 11.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGRB_0 compare match, are performed for channel 0 to 2 TCNT counters, and the data set in TGRB_0 is used as the PWM cycle.

For details of PWM modes, see section 11.4.5, PWM Modes.

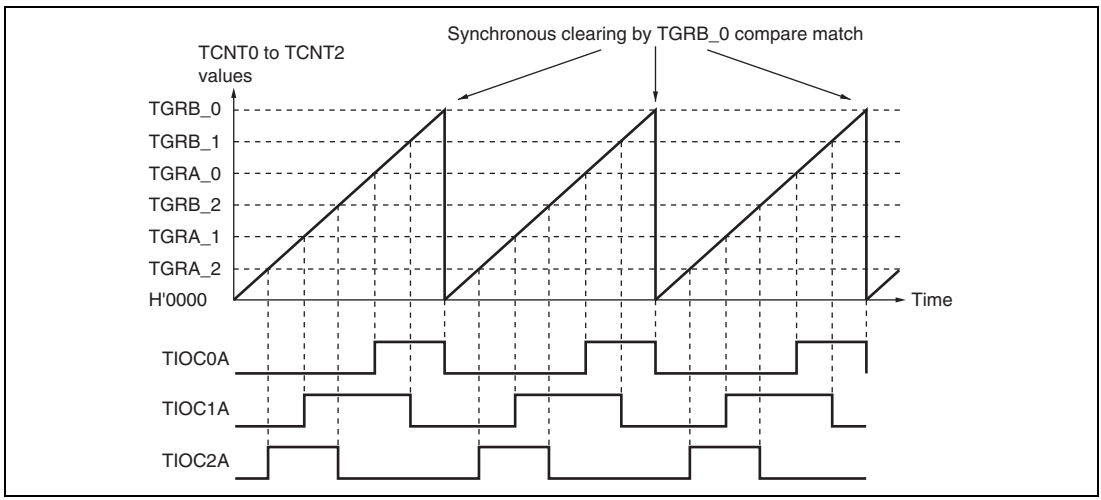


Figure 11.13 Example of Synchronous Operation

11.4.3 Buffer Operation

Buffer operation, provided for channels 0, 3, and 4 enables TGRC and TGRD to be used as buffer registers. In channel 0, 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: TGRE_0 cannot be designated as an input capture register and can only operate as a compare match register.

Table 11.43 shows the register combinations used in buffer operation.

Table 11.43 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

- 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 11.14.

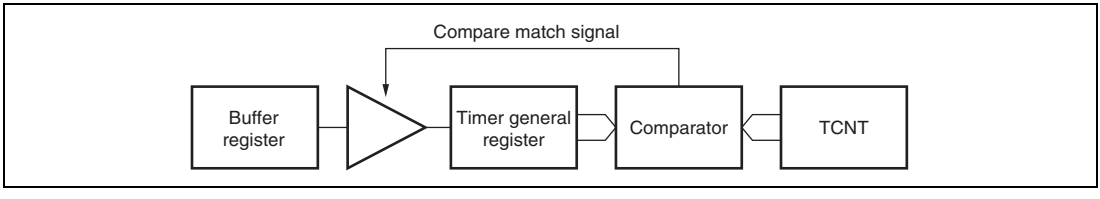


Figure 11.14 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 11.15.

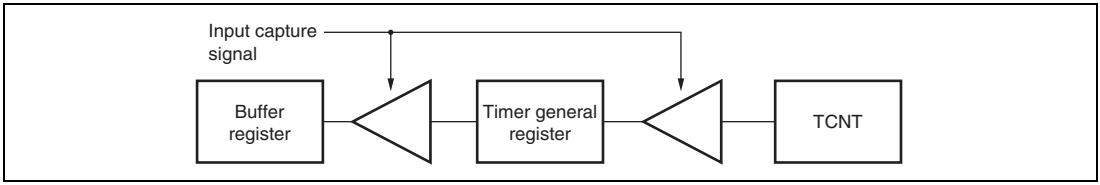


Figure 11.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 11.16 shows an example of the buffer operation setting procedure.

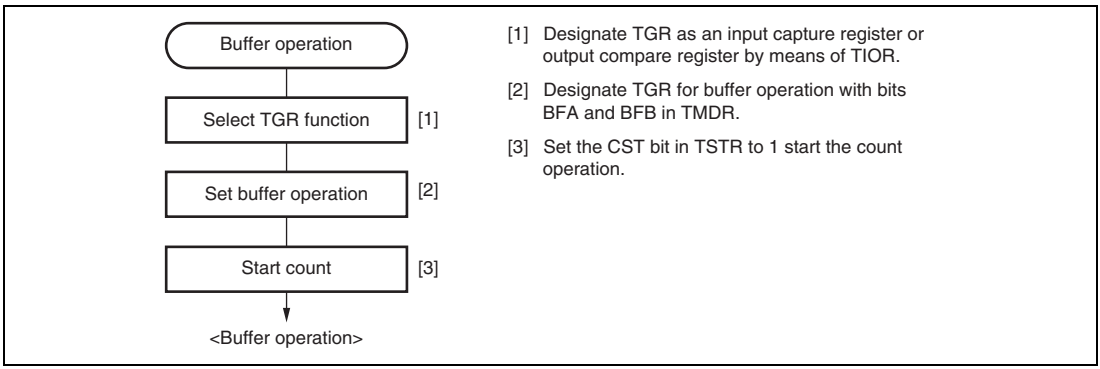


Figure 11.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 11.17 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, 1 output at compare match A, and 0 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 that compare match A occurs.

For details of PWM modes, see section 11.4.5, PWM Modes.

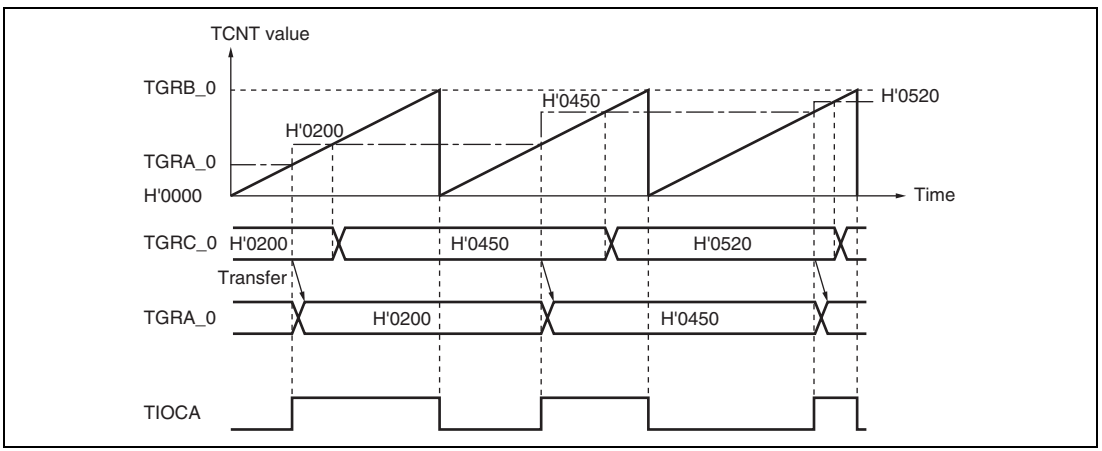


Figure 11.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 11.18 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon the occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

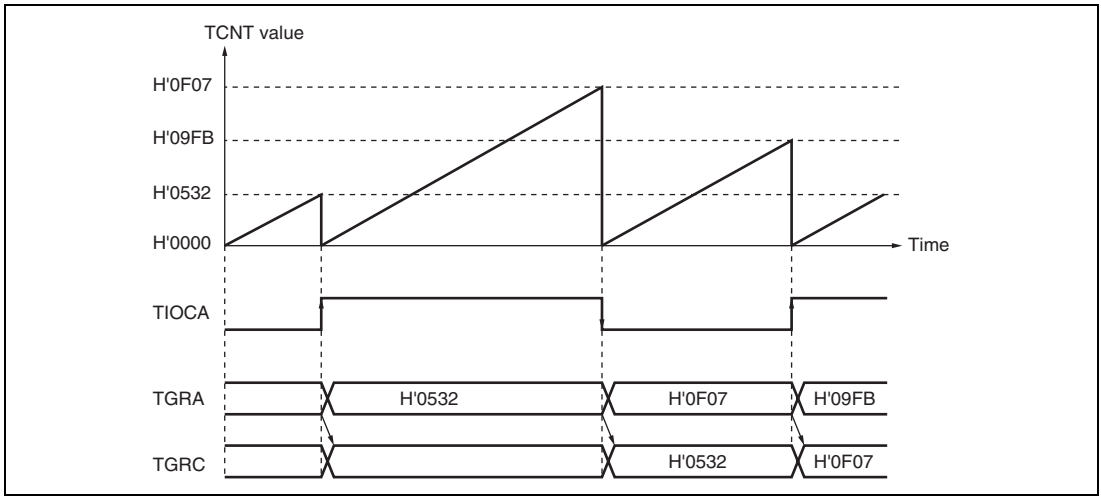


Figure 11.18 Example of Buffer Operation (2)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for channel 0 or in PWM mode 1 for channels 3 and 4 by setting the buffer operation transfer mode registers (TBTM_0, TBTM_3, and TBTM_4). Either compare match (initial setting) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (H'FFFF to H'0000)
- When H'0000 is written to TCNT during counting
- When TCNT is cleared to H'0000 under the condition specified in the CCLR2 to CCLR0 bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 11.19 shows an operation example in which PWM mode 1 is designated for channel 0 and buffer operation is designated for TGRA_0 and TGRC_0. The settings used in this example are TCNT_0 clearing by compare match B, 1 output at compare match A, and 0 output at compare match B. The TTSA bit in TBTM_0 is set to 1.

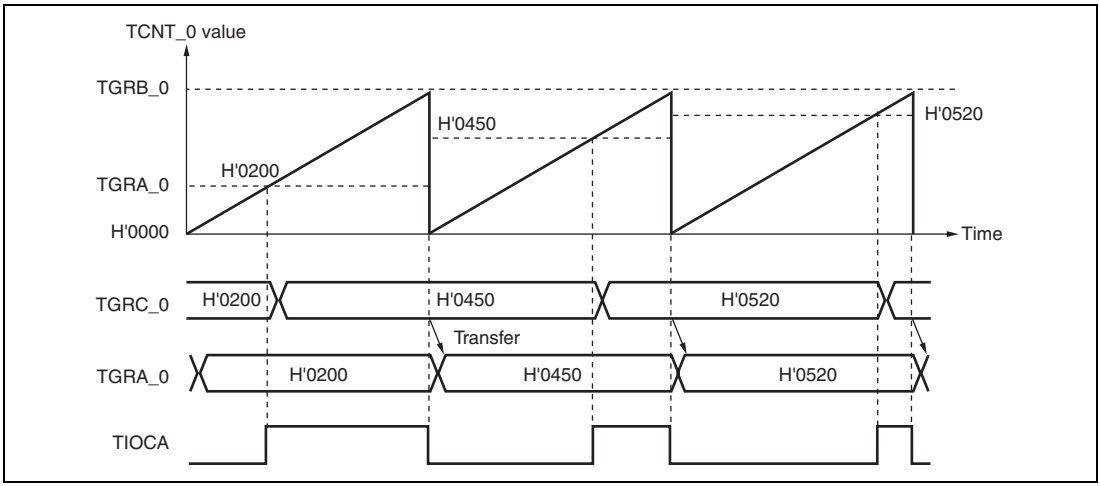


Figure 11.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

11.4.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 counter clock upon overflow/underflow of TCNT_2 as set in bits TPSC0 to TPSC2 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase counting mode.

Table 11.44 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1, the counter clock setting is invalid and the counters operates independently in phase counting mode.

Table 11.44 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2

For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). Edge detection as the condition for input capture is the detection of edges in the signal produced by taking the logical OR of the signals on the main and additional pins. For details, refer to (4),

Cascaded Operation Example (c). For input capture in cascade connection, refer to section 11.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 11.45 show the TICCR setting and input capture input pins.

Table 11.45 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pins
Input capture from TCNT_1 to TGRA_1	I2AE bit = 0 (initial value)	TIOC1A
	I2AE bit = 1	TIOC1A, TIOC2A
Input capture from TCNT_1 to TGRB_1	I2BE bit = 0 (initial value)	TIOC1B
	I2BE bit = 1	TIOC1B, TIOC2B
Input capture from TCNT_2 to TGRA_2	I1AE bit = 0 (initial value)	TIOC2A
	I1AE bit = 1	TIOC2A, TIOC1A
Input capture from TCNT_2 to TGRB_2	I1BE bit = 0 (initial value)	TIOC2B
	I1BE bit = 1	TIOC2B, TIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 11.20 shows an example of the setting procedure for cascaded operation.

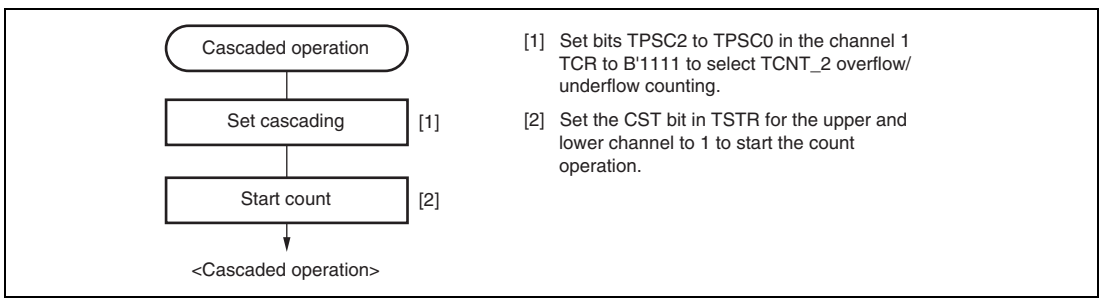


Figure 11.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 11.21 illustrates the operation when TCNT_2 overflow/underflow counting has been set for TCNT_1 and phase counting mode has been designated for channel 2.

TCNT_1 is incremented by TCNT_2 overflow and decremented by TCNT_2 underflow.

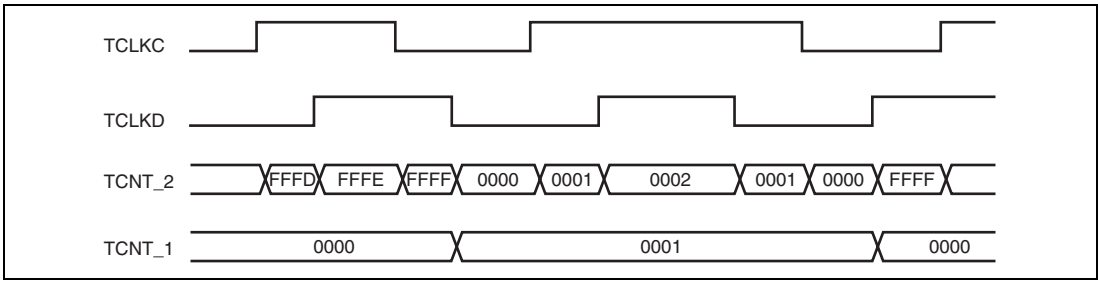


Figure 11.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 11.22 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected the TIOC1A rising edge for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both TIOC1A and TIOC2A is used for the TGRA_1 input capture condition. For the TGRA_2 input capture condition, the TIOC2A rising edge is used.

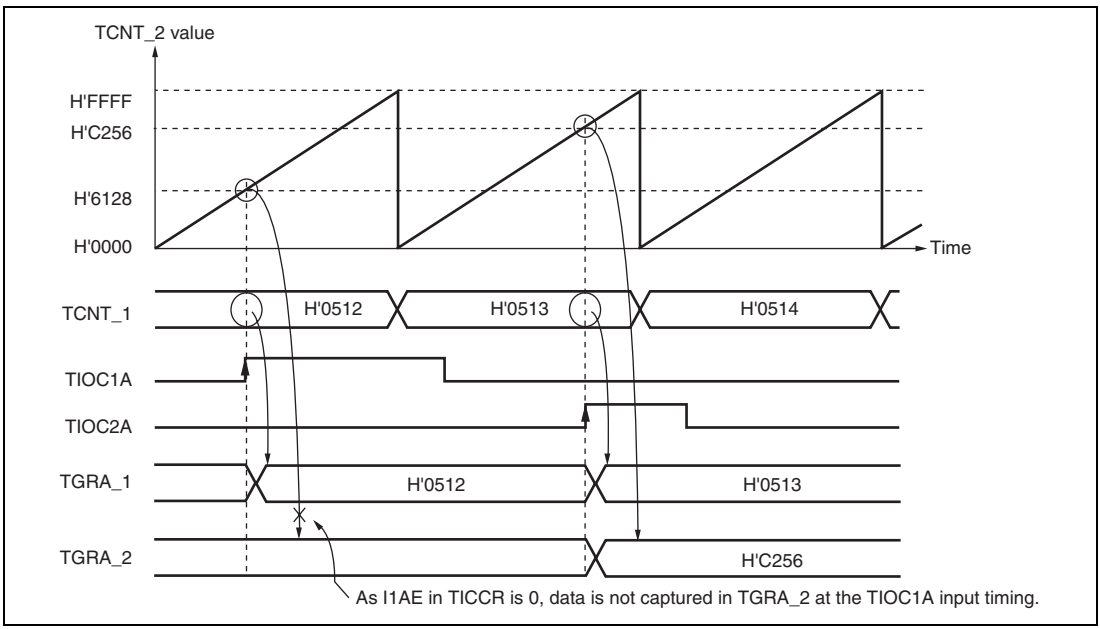
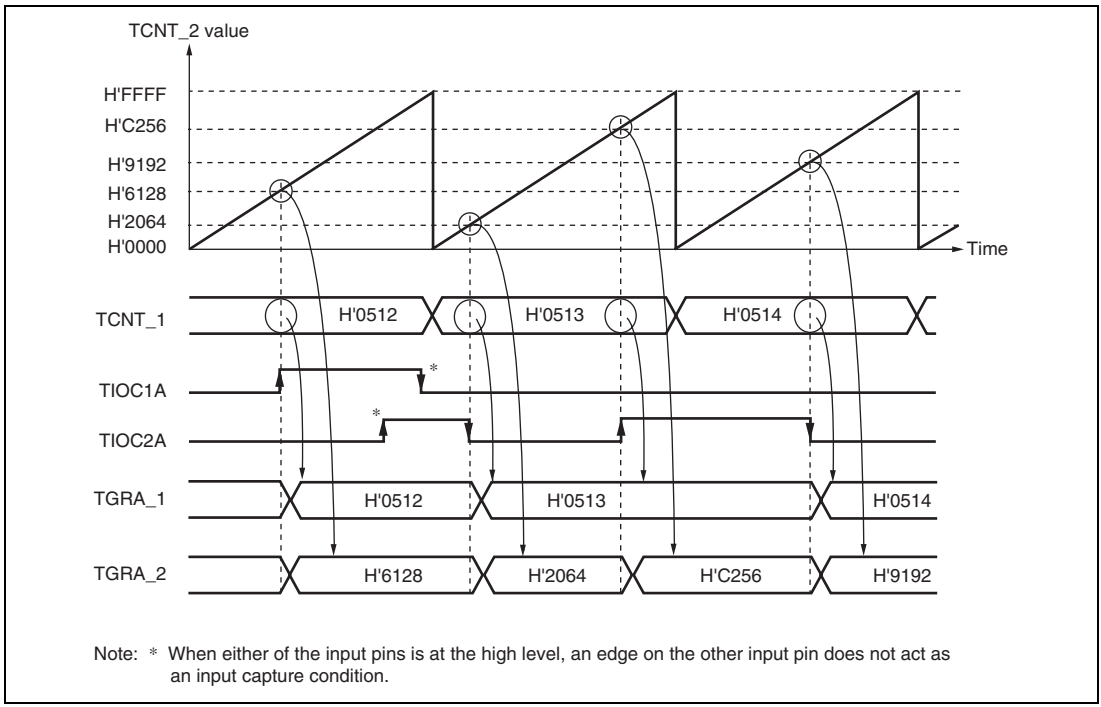


Figure 11.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 11.23 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE and I1AE bits in TICCR have been set to 1 to include the TIOC2A and TIOC1A pins in the TGRA_1 and TGRA_2 input capture conditions, respectively. In this example, the IOA0 to IOA3 bits in both TIOR_1 and TIOR_2 have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of TIOC1A and TIOC2A input is used for the TGRA_1 and TGRA_2 input capture conditions.



Note: * When either of the input pins is at the high level, an edge on the other input pin does not act as an input capture condition.

Figure 11.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 11.24 illustrates the operation when TCNT_1 and TCNT_2 have been cascaded and the I2AE bit in TICCR has been set to 1 to include the TIOC2A pin in the TGRA_1 input capture conditions. In this example, the IOA0 to IOA3 bits in TIOR_1 have selected TGRA_0 compare match or input capture occurrence for the input capture timing while the IOA0 to IOA3 bits in TIOR_2 have selected the TIOC2A rising edge for the input capture timing.

Under these conditions, as TIOR_1 has selected TGRA_0 compare match or input capture occurrence for the input capture timing, the TIOC2A edge is not used for TGRA_1 input capture condition although the I2AE bit in TICCR has been set to 1.

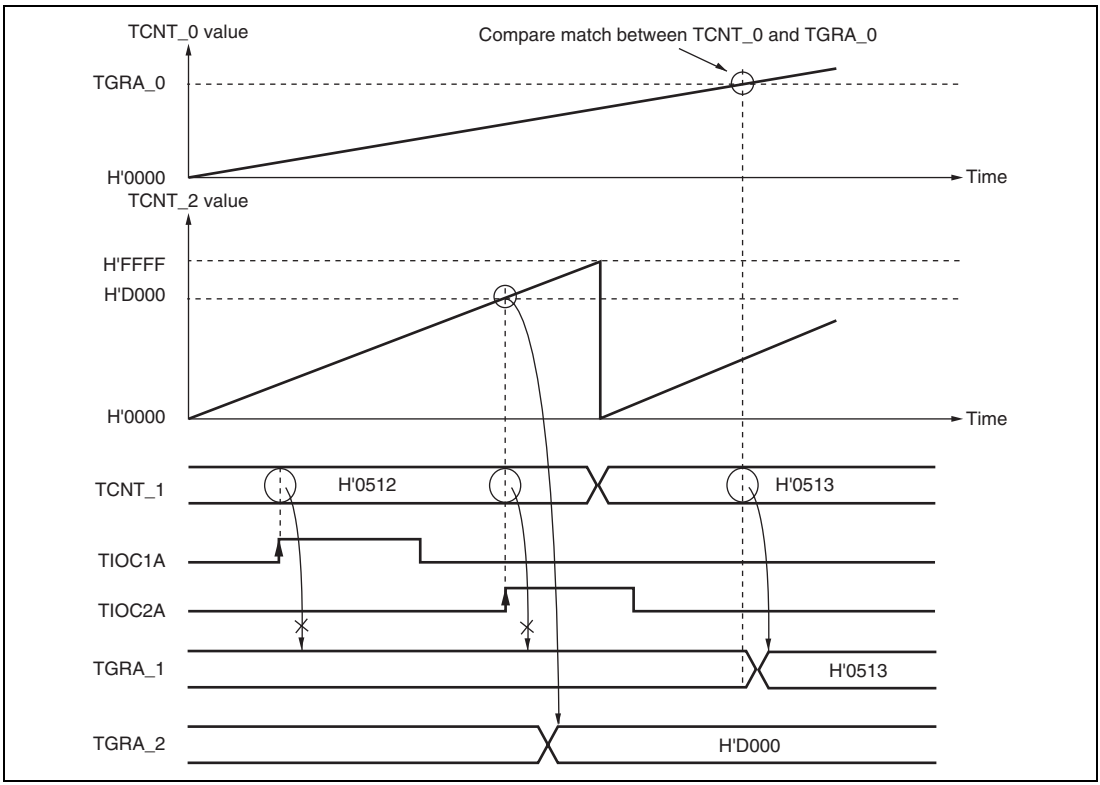


Figure 11.24 Cascaded Operation Example (d)

11.4.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. The output level can be selected as 0, 1, or toggle output in response to a compare match of each TGR.

TGR registers settings can be used to output a PWM waveform in the range of 0% to 100% duty.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA0 to IOA3 and IOC0 to IOC3 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB0 to IOB3 and IOD0 to IOD3 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by the cycle register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 8-phase PWM output is possible in combination use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 11.46.

Table 11.46 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGRA_0	TIOC0A	TIOC0A
	TGRB_0		TIOC0B
	TGRC_0	TIOC0C	TIOC0C
	TGRD_0		TIOC0D
1	TGRA_1	TIOC1A	TIOC1A
	TGRB_1		TIOC1B
2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	Cannot be set
	TGRB_3		Cannot be set
	TGRC_3	TIOC3C	Cannot be set
	TGRD_3		Cannot be set
4	TGRA_4	TIOC4A	Cannot be set
	TGRB_4		Cannot be set
	TGRC_4	TIOC4C	Cannot be set
	TGRD_4		Cannot be set

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

(1) Example of PWM Mode Setting Procedure

Figure 11.25 shows an example of the PWM mode setting procedure.

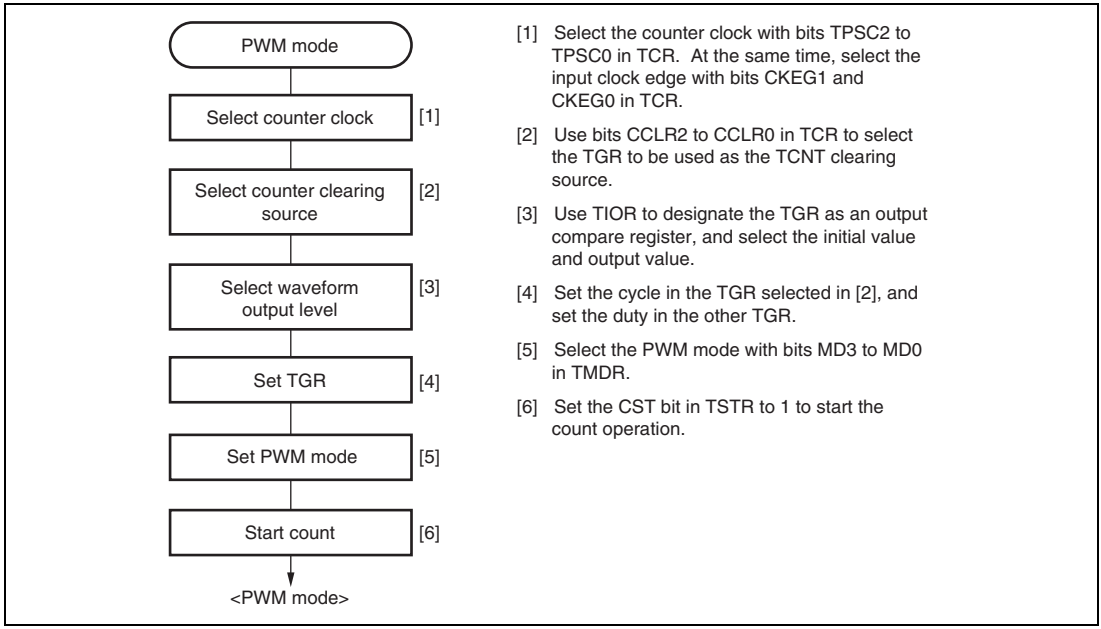


Figure 11.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 11.26 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in the TGRB registers are used as the duty levels.

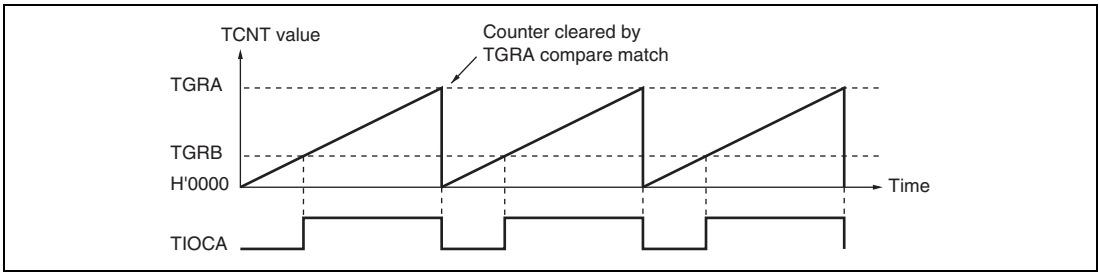


Figure 11.26 Example of PWM Mode Operation (1)

Figure 11.27 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA_0 to TGRD_0, TGRA_1), outputting a 5-phase PWM waveform.

In this case, the value set in TGRB_1 is used as the cycle, and the values set in the other TGRs are used as the duty levels.

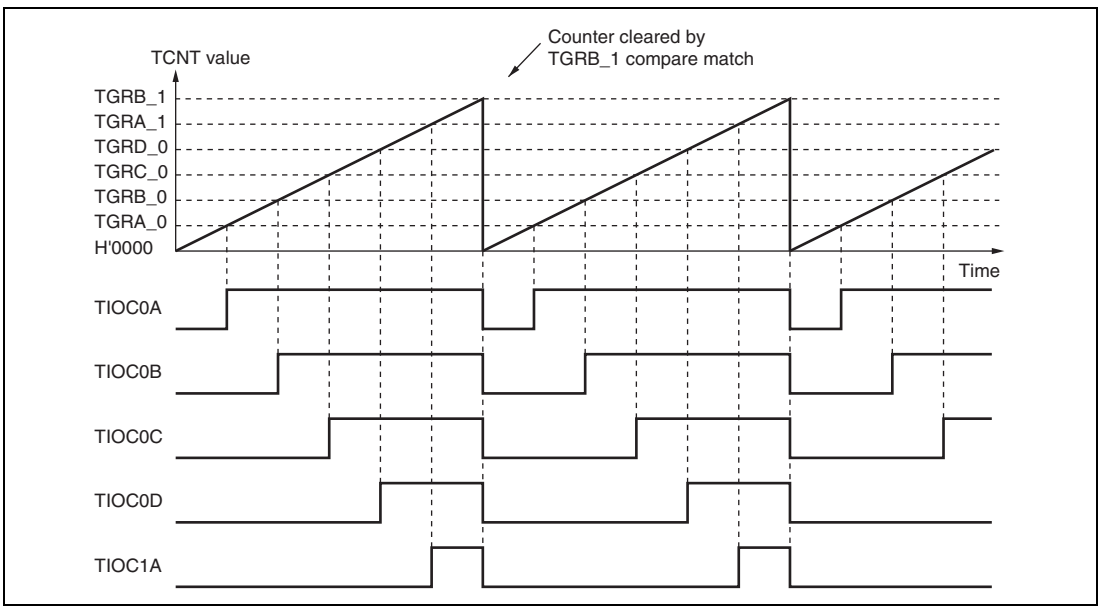


Figure 11.27 Example of PWM Mode Operation (2)

Figure 11.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

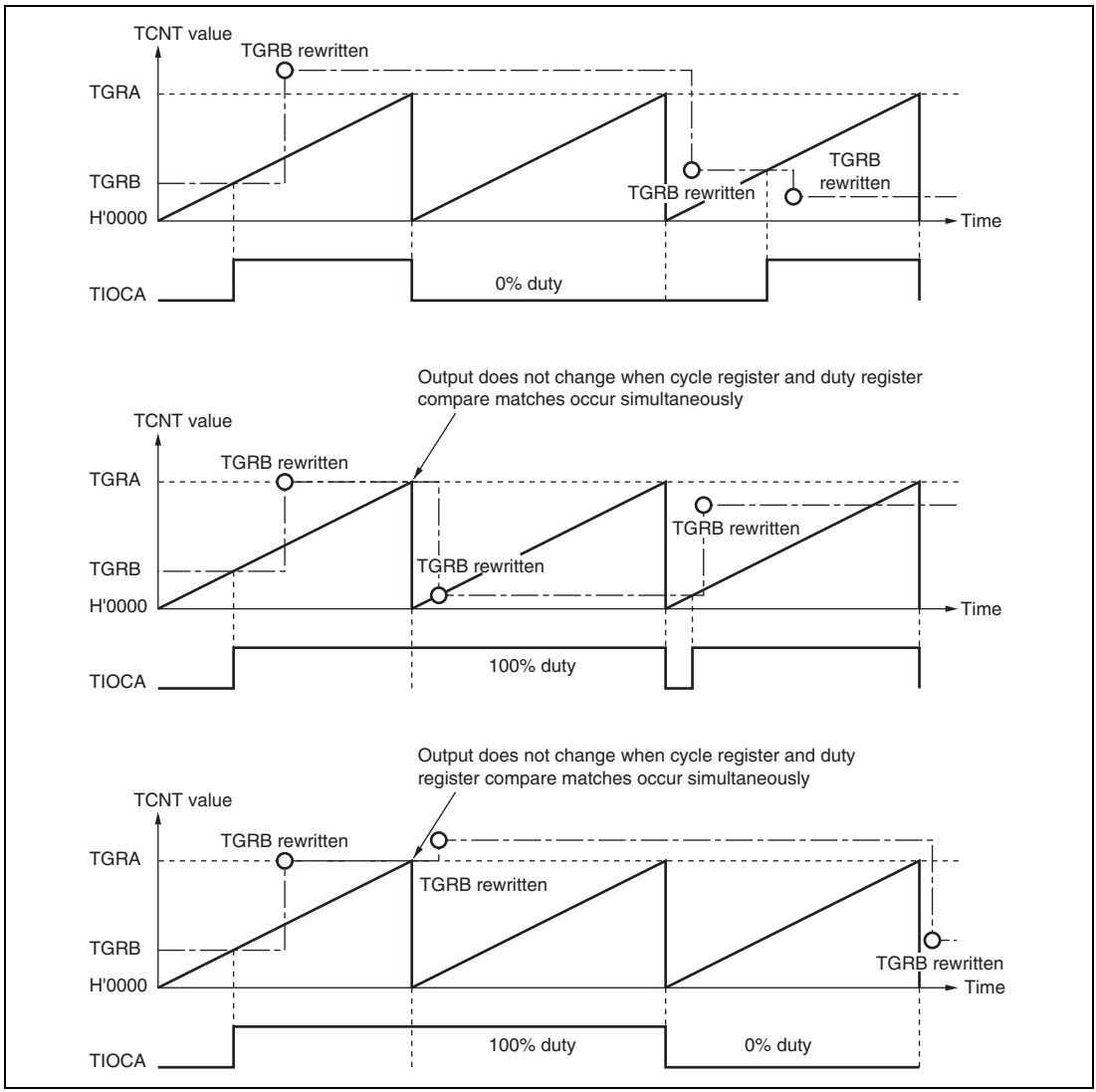


Figure 11.28 Example of PWM Mode Operation (3)

11.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1 and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC0 to TPSC2 and bits CKEG0 and CKEG1 in TCR. However, the functions of bits CCLR0 and CCLR1 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 overflow occurs when TCNT is counting up, the TCFV flag in TSR is set; if underflow occurs when TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag reveals whether TCNT is counting up or down.

Table 11.47 shows the correspondence between external clock pins and channels.

Table 11.47 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 11.29 shows an example of the phase counting mode setting procedure.

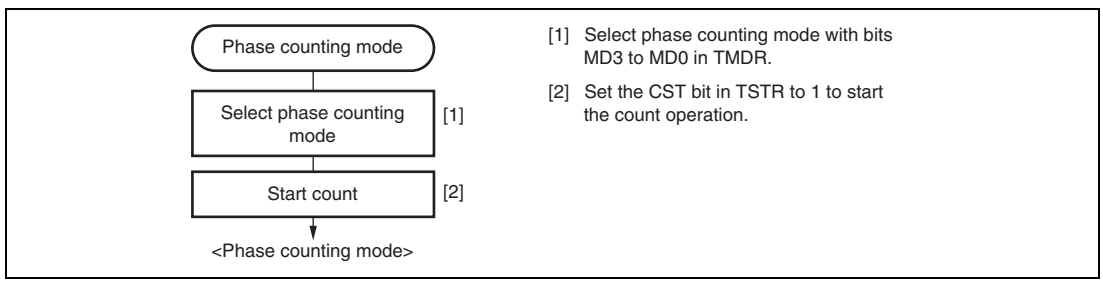


Figure 11.29 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes according to the count conditions.

(a) Phase counting mode 1

Figure 11.30 shows an example of phase counting mode 1 operation, and table 11.48 summarizes the TCNT up/down-count conditions.

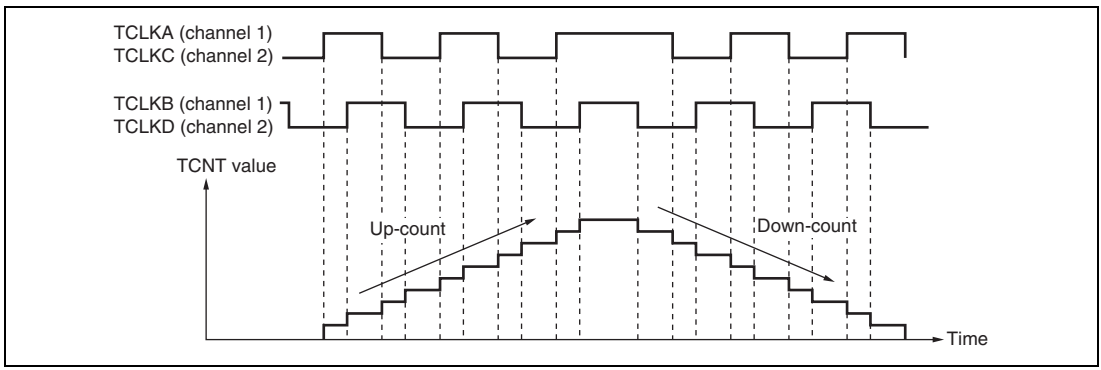


Figure 11.30 Example of Phase Counting Mode 1 Operation

Table 11.48 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	
⌐↓	Low level	

[Legend]

⌐↑: Rising edge

⌐↓: Falling edge

(b) Phase counting mode 2

Figure 11.31 shows an example of phase counting mode 2 operation, and table 11.49 summarizes the TCNT up/down-count conditions.

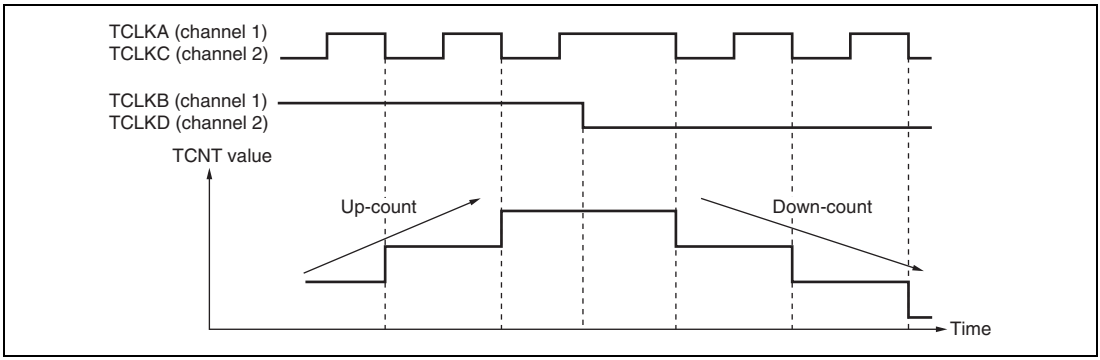


Figure 11.31 Example of Phase Counting Mode 2 Operation

Table 11.49 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

[Legend]
↑: Rising edge
↓: Falling edge

(c) Phase counting mode 3

Figure 11.32 shows an example of phase counting mode 3 operation, and table 11.50 summarizes the TCNT up/down-count conditions.

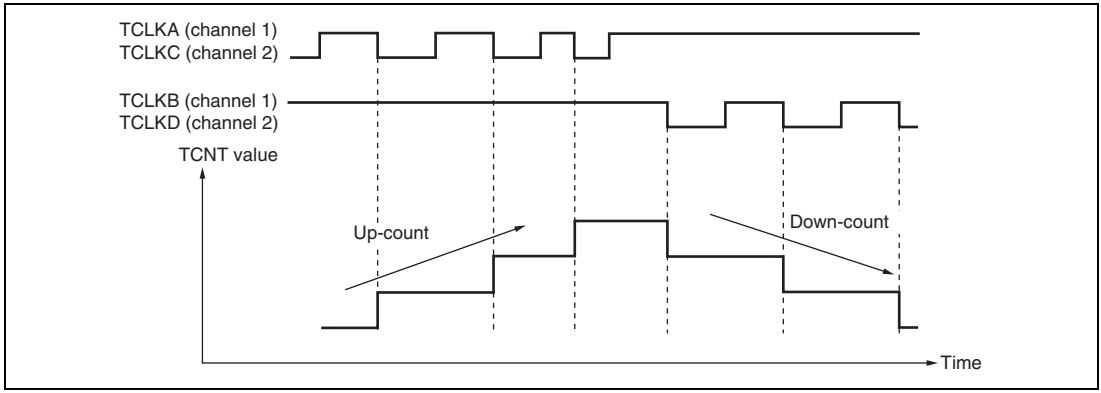


Figure 11.32 Example of Phase Counting Mode 3 Operation

Table 11.50 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

[Legend]
↑: Rising edge
↓: Falling edge

(d) Phase counting mode 4

Figure 11.33 shows an example of phase counting mode 4 operation, and table 11.51 summarizes the TCNT up/down-count conditions.

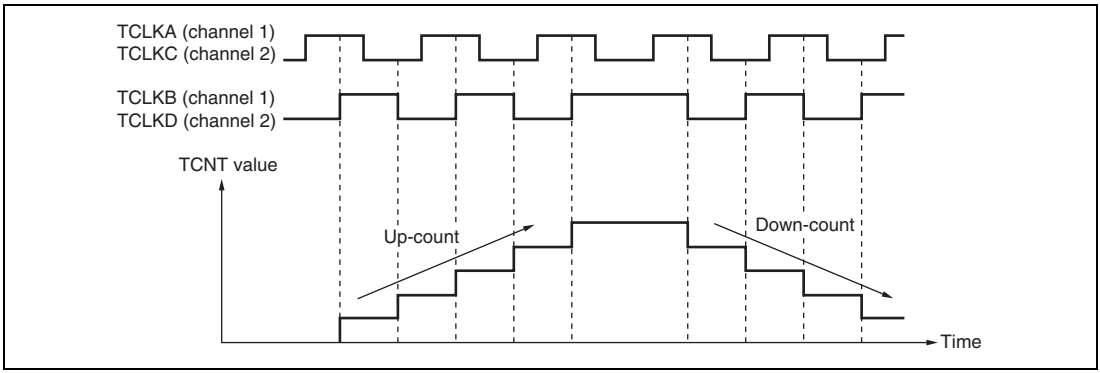


Figure 11.33 Example of Phase Counting Mode 4 Operation

Table 11.51 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

[Legend]
↑: Rising edge
↓: Falling edge

(3) Phase Counting Mode Application Example

Figure 11.34 shows an example in which channel 1 is in phase counting mode, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGRC_0 compare match; TGRA_0 and TGRC_0 are used for the compare match function and are set with the speed control period and position control period. TGRB_0 is used for input capture, with TGRB_0 and TGRD_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB_0 input capture source, and the pulse widths of 2-phase encoder 4-multiplication pulses are detected.

TGRA_1 and TGRB_1 for channel 1 are designated for input capture, and channel 0 TGRA_0 and TGRC_0 compare matches are selected as the input capture source and store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

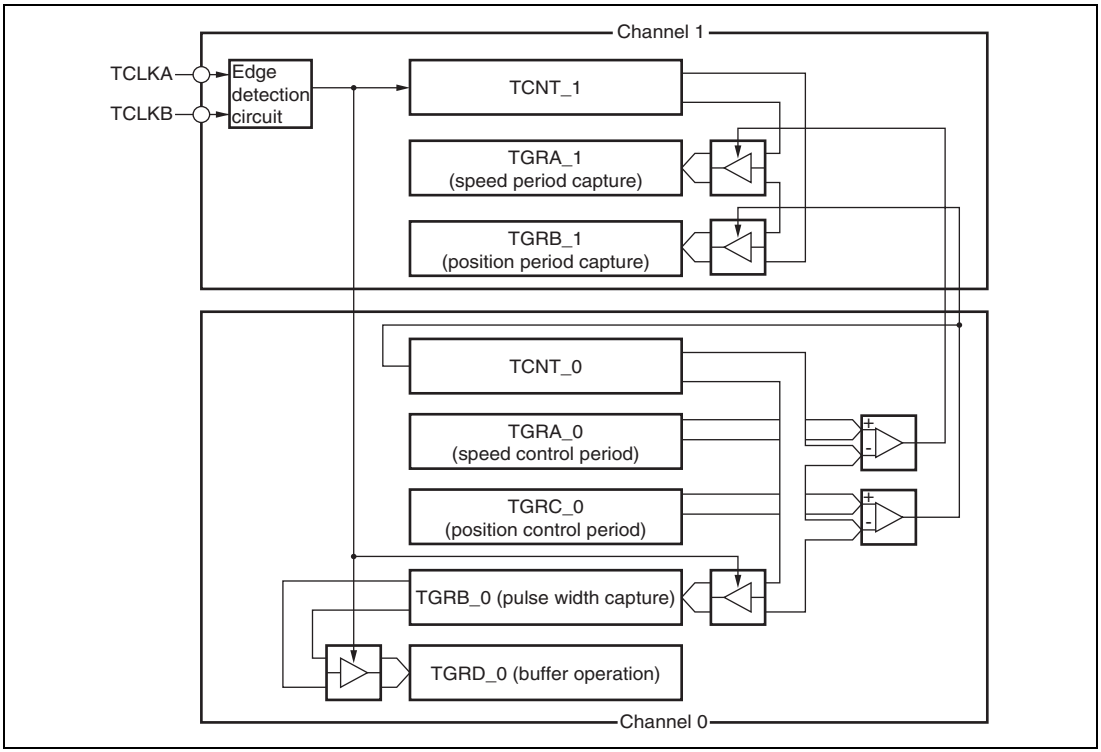


Figure 11.34 Phase Counting Mode Application Example

11.4.7 Reset-Synchronized PWM Mode

In reset-synchronized PWM mode, three-phase output of positive and negative PWM waveforms that share a common wave transition point can be obtained by combining channels 3 and 4.

When set for reset-synchronized PWM mode, the TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, and TIOC4D pins function as PWM output pins and TCNT3 functions as an upcounter.

Table 11.52 shows the PWM output pins used. Table 11.53 shows the settings of the registers.

Table 11.52 Output Pins for Reset-Synchronized PWM Mode

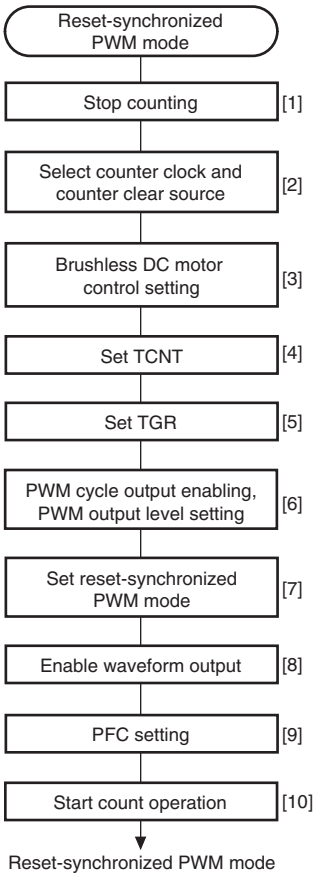
Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 11.53 Register Settings for Reset-Synchronized PWM Mode

Register	Description of Setting
TCNT_3	Initial setting of H'0000
TCNT_4	Initial setting of H'0000
TGRA_3	Set count cycle for TCNT_3
TGRB_3	Sets the turning point for PWM waveform output by the TIOC3B and TIOC3D pins
TGRA_4	Sets the turning point for PWM waveform output by the TIOC4A and TIOC4C pins
TGRB_4	Sets the turning point for PWM waveform output by the TIOC4B and TIOC4D pins

(1) Procedure for Selecting the Reset-Synchronized PWM Mode

Figure 11.35 shows an example of procedure for selecting reset-synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in the TSTR to 0 to halt the counting of TCNT. The reset-synchronized PWM mode must be set up while TCNT_3 and TCNT_4 are halted.

[2] Set bits TPSC2-TPSC0 and CKEG1 and CKEG0 in the TCR_3 to select the counter clock and clock edge for channel 3. Set bits CCLR2-CCLR0 in the TCR_3 to select TGRA compare-match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Reset TCNT_3 and TCNT_4 to H'0000.

[5] TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.
$X \leq$ TGRA_3 (X: set value).

[6] Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 10.3.

[7] Set bits MD3-MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. Do not set to TMDR_4.

[8] Set the enabling/disabling of the PWM waveform output pin in TOER.

[9] Set the port control register and the port I/O register.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

Figure 11.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Reset-Synchronized PWM Mode Operation

Figure 11.36 shows an example of operation in reset-synchronized PWM mode. TCNT_3 and TCNT_4 operate as upcounters. The counter is cleared when a TCNT_3 and TGRA_3 compare-match occurs, and then begins incrementing from H'0000. The PWM output pin output toggles with each occurrence of a TGRB_3, TGRA_4, TGRB_4 compare-match, and upon counter clears.

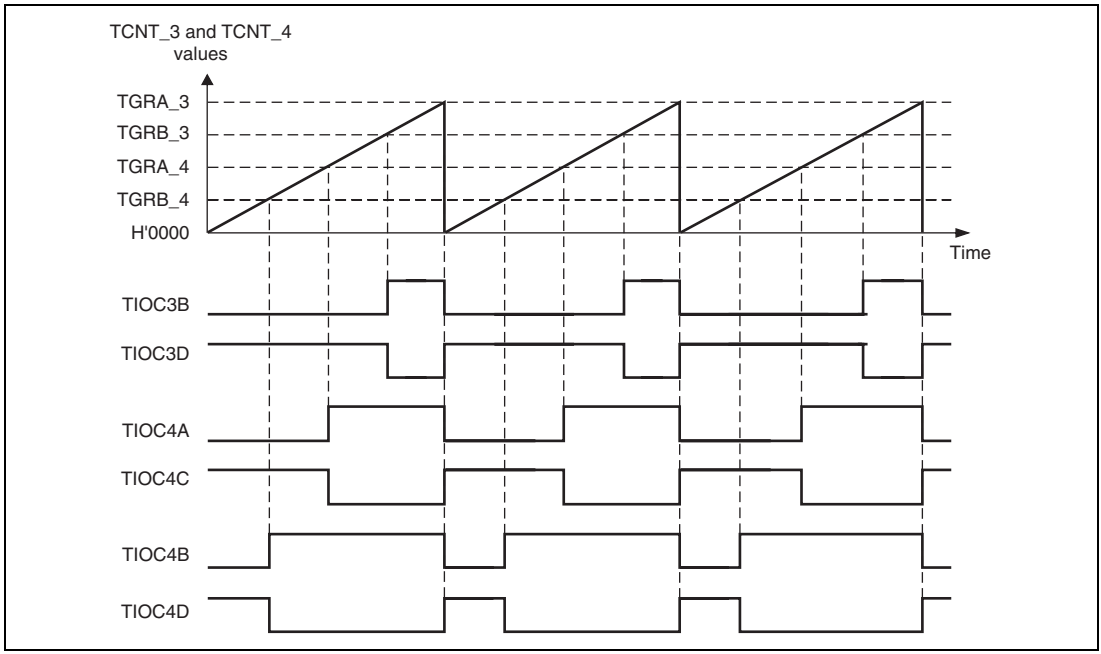


Figure 11.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

11.4.8 Complementary PWM Mode

In complementary PWM mode, three-phase output of non-overlapping positive and negative PWM waveforms can be obtained by combining channels 3 and 4. PWM waveforms without non-overlapping interval are also available.

In complementary PWM mode, TIOC3B, TIOC3D, TIOC4A, TIOC4B, TIOC4C, and TIOC4D pins function as PWM output pins, the TIOC3A pin can be set for toggle output synchronized with the PWM period. TCNT_3 and TCNT_4 function as up/down counters.

Table 11.54 shows the PWM output pins used. Table 11.55 shows the settings of the registers used. Figure 11.37 describes a block diagram of channels 3 and 4 in complementary PWM mode.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 11.54 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
3	TIOC3A	Toggle output synchronized with PWM period (or I/O port)
	TIOC3B	PWM output pin 1
	TIOC3C	I/O port*
	TIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
4	TIOC4A	PWM output pin 2
	TIOC4B	PWM output pin 3
	TIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	TIOC4D	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 TIOC3C pin as a timer I/O pin in complementary PWM mode.

Table 11.55 Register Settings for Complementary PWM Mode

Channel	Counter/Register	Description	Read/Write from CPU
3	TCNT_3	Start of up-count from value set in dead time register	Maskable by TRWER setting*
	TGRA_3	Set TCNT_3 upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*
	TGRB_3	PWM output 1 compare register	Maskable by TRWER setting*
	TGRC_3	TGRA_3 buffer register	Always readable/writable
	TGRD_3	PWM output 1/TGRB_3 buffer register	Always readable/writable
4	TCNT_4	Up-count start, initialized to H'0000	Maskable by TRWER setting*
	TGRA_4	PWM output 2 compare register	Maskable by TRWER setting*
	TGRB_4	PWM output 3 compare register	Maskable by TRWER setting*
	TGRC_4	PWM output 2/TGRA_4 buffer register	Always readable/writable
	TGRD_4	PWM output 3/TGRB_4 buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set TCNT_4 and TCNT_3 offset value (dead time value)	Maskable by TRWER setting*
Timer cycle data register (TCDR)		Set TCNT_4 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/TGRB_3 temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/TGRA_4 temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/TGRB_4 temporary register	Not readable/writable

Note: * Access can be enabled or disabled according to the setting of bit 0 (RWE) in TRWER (timer read/write enable register).

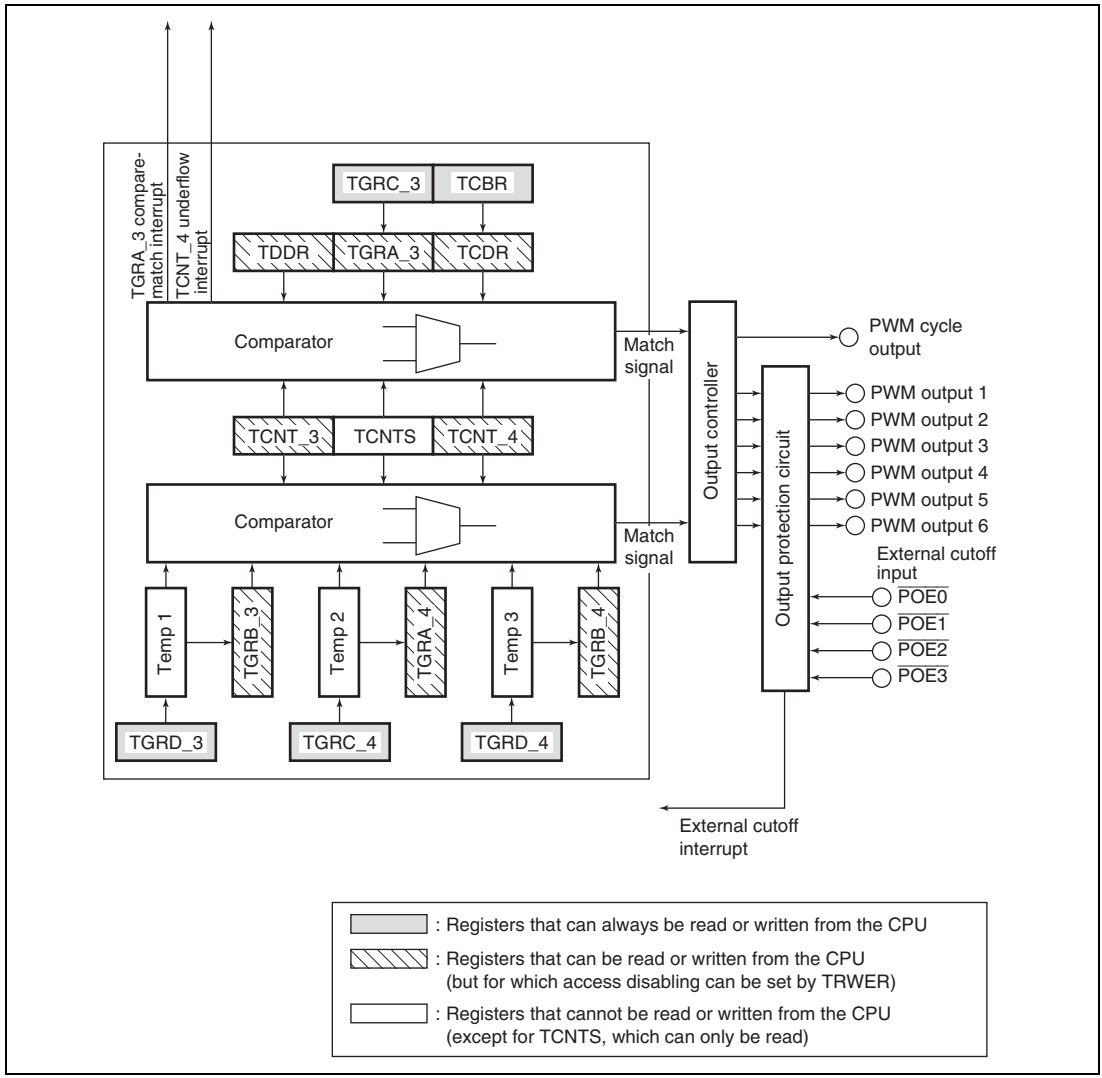
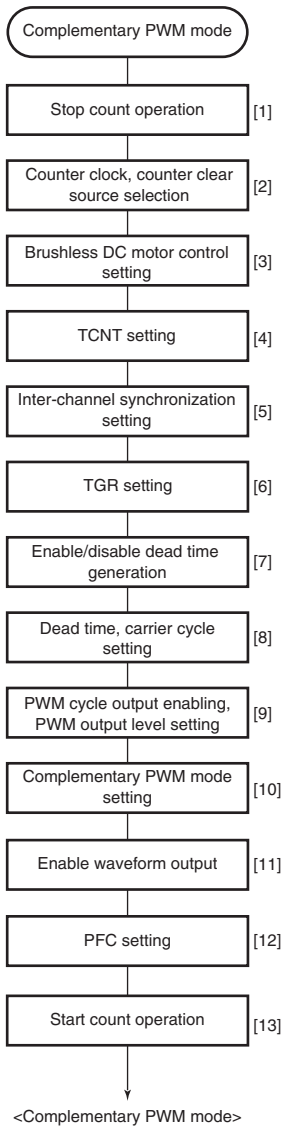


Figure 11.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 11.38.



[1] Clear bits CST3 and CST4 in the timer start register (TSTR) to 0, and halt timer counter (TCNT) operation. Perform complementary PWM mode setting when TCNT_3 and TCNT_4 are stopped.

[2] Set the same counter clock and clock edge for channels 3 and 4 with bits TPSC2-TPSC0 and bits CKEG1 and CKEG0 in the timer control register (TCR). Use bits CCLR2-CCLR0 to set synchronous clearing only when restarting by a synchronous clear from another channel during complementary PWM mode operation.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in TCNT_3. Set TCNT_4 to H'0000.

[5] Set only when restarting by a synchronous clear from another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with channels 3 and 4 using the timer synchro register (TSYR).

[6] Set the output PWM duty in the duty registers (TGRB_3, TGRA_4, TGRB_4) and buffer registers (TGRD_3, TGRC_4, TGRD_4). Set the same initial value in each corresponding TGR.

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the timer dead time enable register (TDER) so that no dead time is generated.

[8] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the timer cycle data register (TCDR) and timer cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in TGRA_3 and TGRC_3.

[9] Select enabling/disabling of toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 10.3.

[10] Select complementary PWM mode in timer mode register 3 (TMDR_3). Do not set in TMDR_4.

[11] Set enabling/disabling of PWM waveform output pin output in the timer output master enable register (TOER).

[12] Set the port control register and the port I/O register.

[13] Set bits CST3 and CST4 in TSTR to 1 simultaneously to start the count operation.

Figure 11.38 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 11.39 illustrates counter operation in complementary PWM mode, and figure 11.40 shows an example of complementary PWM mode operation.

(a) Counter Operation

In complementary PWM mode, three counters—TCNT_3, TCNT_4, and TCNTS—perform up/down-count operations.

TCNT_3 is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST bit is set to 1, TCNT_3 counts up to the value set in TGRA_3, then switches to down-counting when it matches TGRA_3. When the TCNT3 value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

TCNT_4 is initialized to H'0000.

When the CST bit is set to 1, TCNT4 counts up in synchronization with TCNT_3, and switches to down-counting when it matches TCDR. On reaching H'0000, TCNT4 switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It need not be initialized.

When TCNT_3 matches TCDR during TCNT_3 and TCNT_4 up/down-counting, down-counting is started, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches TGRA_3, it is cleared to H'0000.

When TCNT_4 matches TDDR during TCNT_3 and TCNT_4 down-counting, up-counting is started, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches H'0000, it is set with the value in TGRA_3.

TCNTS is compared with the compare register and temporary register in which the PWM duty is set during the count operation only.

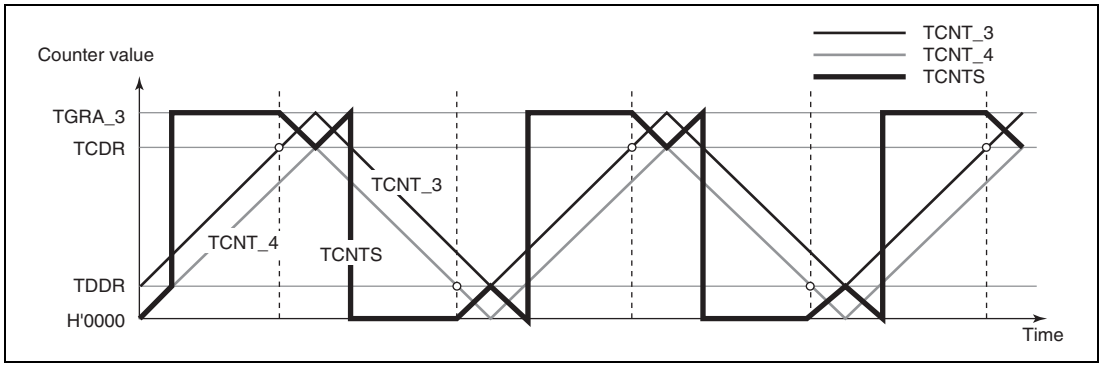


Figure 11.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 11.40 shows an example of complementary PWM mode operation.

The registers which are constantly compared with the counters to perform PWM output are TGRB_3, TGRA_4, and TGRB_4. When these registers match the counter, the value set in bits OLSN and OLSP in the timer output control register (TOCR) is output.

The buffer registers for these compare registers are TGRD_3, TGRC_4, and TGRD_4.

Between a buffer register and compare register there is a temporary register. The temporary registers cannot be accessed by the CPU.

Data in a compare register is changed by writing the 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 TGRA_3 when counting up, or H'0000 when counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD3 to MD0 in the timer mode register (TMDR). Figure 11.40 shows an example in which the mode is selected in which the change is made in the trough.

In the Tb interval (tb1 in figure 11.40) in which data transfer to the temporary register is not performed, 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-phase output, with the compare register containing the pre-change data, and the temporary register containing the new data. In this interval, the three counters—TCNT_3, TCNT_4, and TCNTS—and two registers—compare register and temporary register—are compared, and PWM output controlled accordingly.

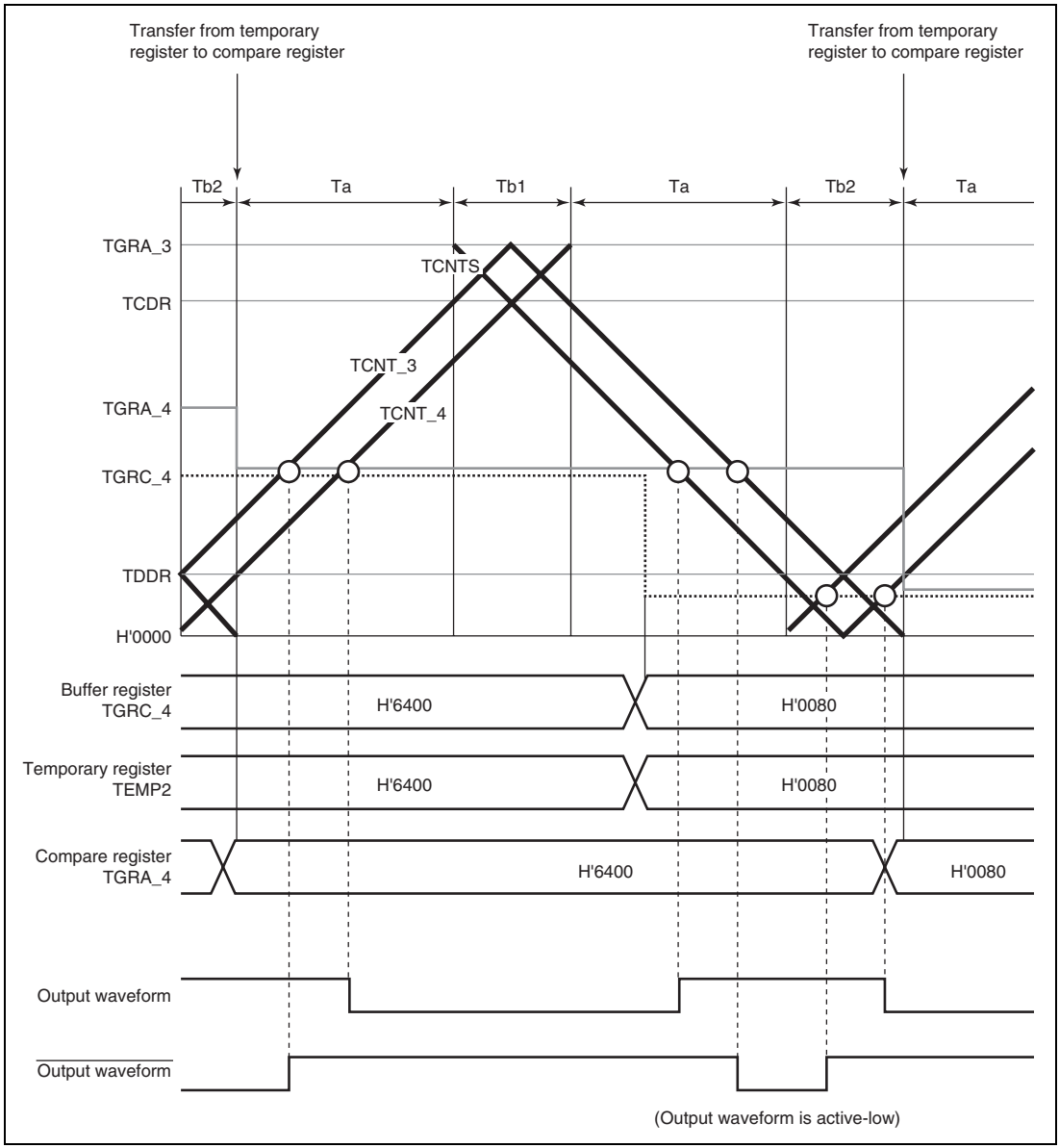


Figure 11.40 Example of Complementary PWM Mode Operation

(c) Initialization

In complementary PWM mode, there are six registers that must be initialized. 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 MD3 to MD0 in the timer mode register (TMDR), the following initial register values must be set.

TGRC_3 operates as the buffer register for TGRA_3, and should be set with 1/2 the PWM carrier cycle + dead time Td. The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time Td in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, TGRC_3 and TGRA_3 should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1.

Set the respective initial PWM duty values in buffer registers TGRD_3, TGRC_4, and TGRD_4.

The values set in the five buffer registers excluding TDDR are transferred simultaneously to the corresponding compare registers when complementary PWM mode is set.

Set TCNT_4 to H'0000 before setting complementary PWM mode.

Table 11.56 Registers and Counters Requiring Initialization

Register/Counter	Set Value
TGRC_3	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM carrier cycle
TGRD_3, TGRC_4, TGRD_4	Initial PWM duty value for each phase
TCNT_4	H'0000

Note: The TGRC_3 set value must be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC_3 must 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 TCNT_3 counter start value, and creates non-overlap between TCNT_3 and TCNT_4. Complementary PWM mode should be cleared before changing the contents of TDDR.

(f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER can be cleared to 0 only when 0 is written to it after reading TDER = 1.

TGRA_3 and TGRC_3 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 11.41 shows an example of operation without dead time.

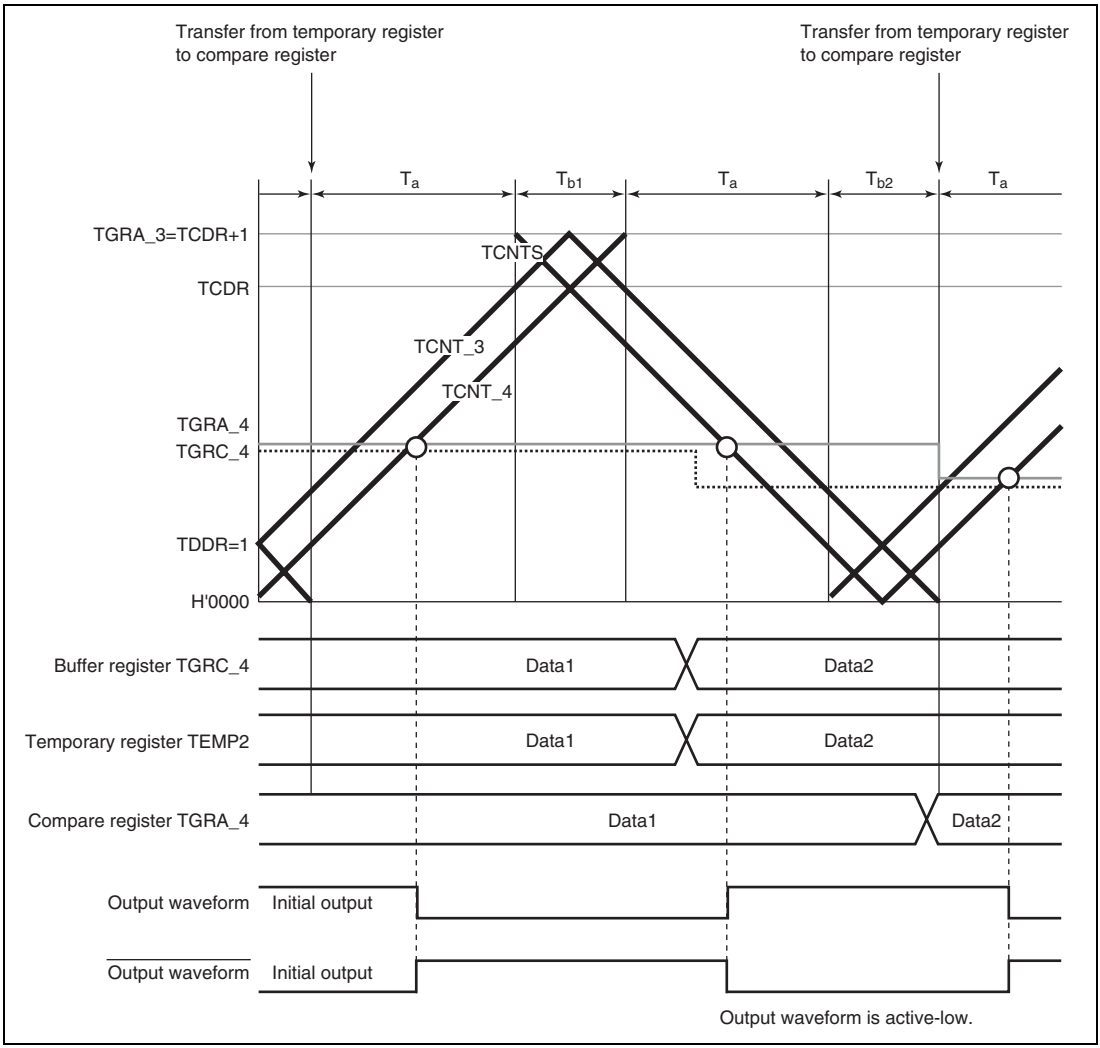


Figure 11.41 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—TGRA_3, in which the TCNT_3 upper limit value is set, and TCDR, in which the TCNT_4 upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: TGRA_3 set value = TCDR set value + TDDR set value
TCDR set value > Double the TDDR set value + 2
Without dead time: TGRA_3 set value = TCDR set value + 1

The TGRA_3 and TCDR settings are made by setting the values in buffer registers TGRC_3 and TCBR. The values set in TGRC_3 and TCBR are transferred simultaneously to TGRA_3 and TCDR in accordance with the transfer timing selected with bits MD3 to MD0 in the timer mode register (TMDR).

The updated PWM cycle is reflected from the next cycle when the data update is performed at the crest, and from the current cycle when performed in the trough. Figure 11.42 illustrates the operation when the PWM cycle is updated at the crest.

See the following section, Register Data Updating, for the method of updating the data in each buffer register.

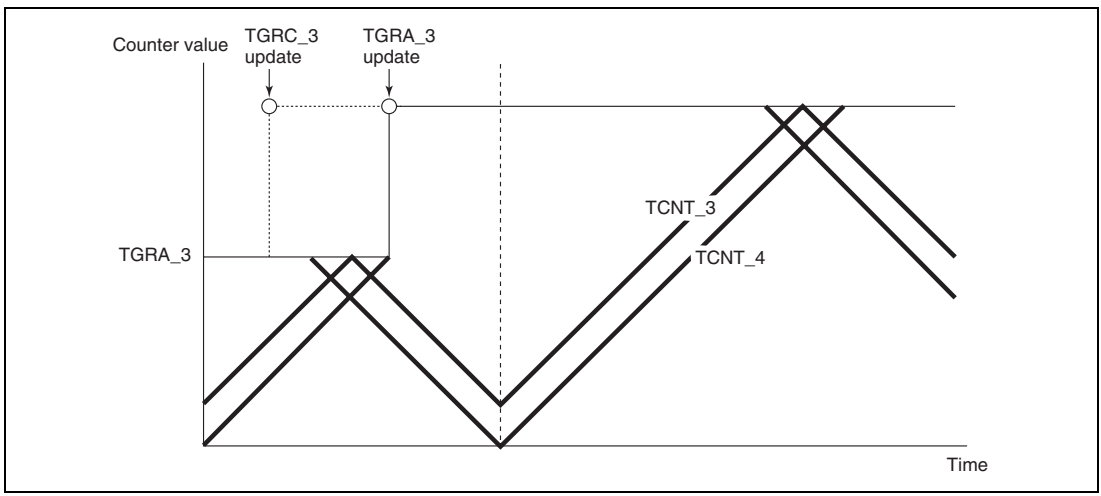


Figure 11.42 Example of PWM Cycle Updating

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five 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. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when 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 MD3 to MD0 in the timer mode register (TMDR). Figure 11.43 shows an example of data updating in complementary PWM mode. This example shows the mode in which data updating is performed at both the counter crest and trough.

When rewriting buffer register data, a write to TGRD_4 must be performed at the end of the update. Data transfer from the buffer registers to the temporary registers is performed simultaneously for all five registers after the write to TGRD_4.

A write to TGRD_4 must be performed after writing data to the registers to be updated, even when not updating all five registers, or when updating the TGRD_4 data. In this case, the data written to TGRD_4 should be the same as the data prior to the write operation.

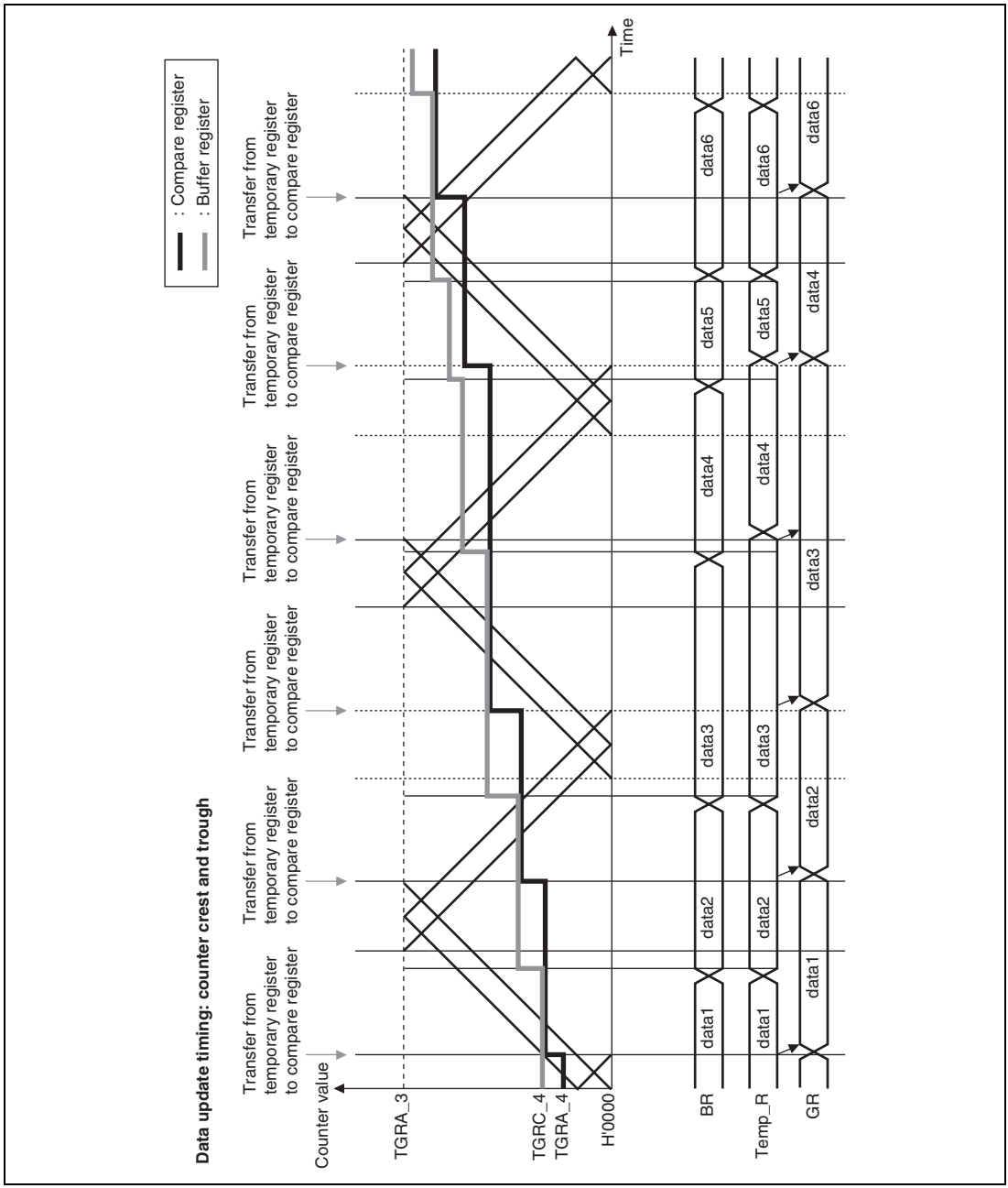


Figure 11.43 Example of Data Update in Complementary PWM Mode

(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 PWM pulse non-active level, and is output from when complementary PWM mode is set with the timer mode register (TMDR) until TCNT_4 exceeds the value set in the dead time register (TDDR). Figure 11.44 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in figure 11.45.

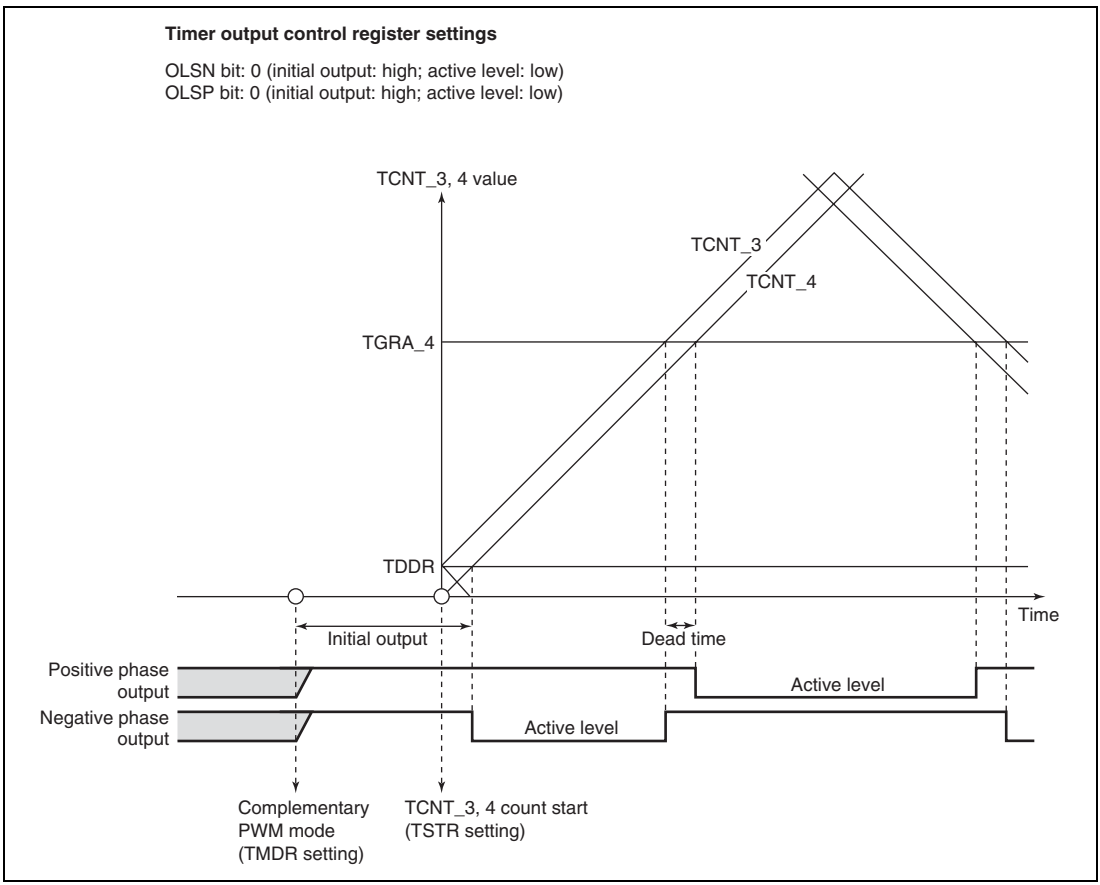


Figure 11.44 Example of Initial Output in Complementary PWM Mode (1)

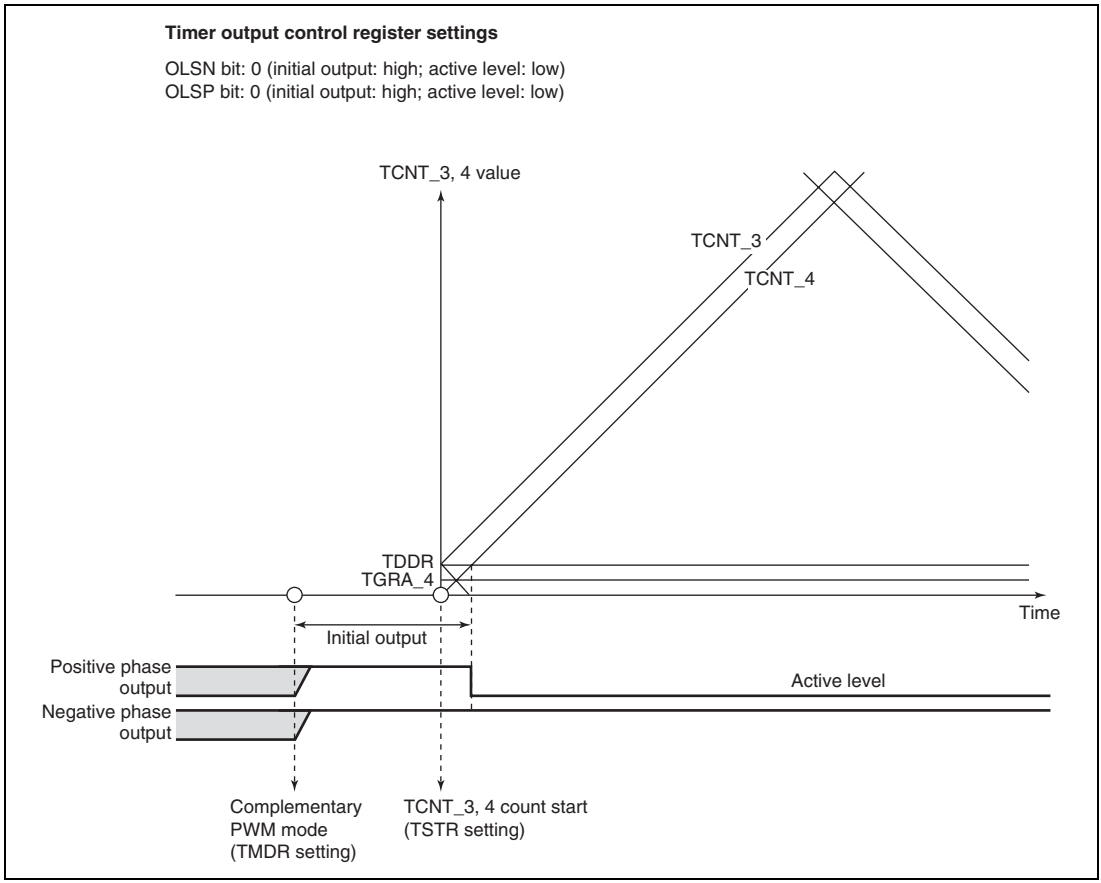


Figure 11.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Complementary PWM Mode PWM Output Generation Method

In complementary PWM mode, 3-phase output is performed of PWM waveforms 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 output level selected in the timer output control register in the event of a compare-match between a counter and compare register. While TCNTS is counting, compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. The relative timing of on and 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 on times do not overlap. Figures 11.46 to 11.48 show examples of waveform generation in complementary PWM mode.

The positive phase/negative phase off timing is generated by a compare-match with the solid-line counter, and the on timing by a compare-match with the dotted-line counter operating 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 occurring prior to **a** are ignored. In the T2 period, compare-match **c** that turns off the positive phase has the highest priority, and compare-matches occurring prior to **c** are ignored.

In normal cases, compare-matches occur in the order **a** → **b** → **c** → **d** (or **c** → **d** → **a'** → **b'**), as shown in figure 11.46.

If compare-matches deviate from the **a** → **b** → **c** → **d** order, since the time for which the negative phase is off is less than twice the dead time, the figure shows the positive phase is not being turned on. If compare-matches deviate from the **c** → **d** → **a'** → **b'** order, since the time for which the positive phase is off is less than twice the dead time, the figure shows the negative phase is not being turned on.

If compare-match **c** occurs first following compare-match **a**, as shown in figure 11.47, compare-match **b** is ignored, and the negative phase is turned on by compare-match **d**. This is because turning off of 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 since the positive phase goes from off to off).

Similarly, in the example in figure 11.48, compare-match **a'** with the new data in the temporary register occurs before compare-match **c**, but other compare-matches occurring up to **c**, which turns off the positive phase, 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.

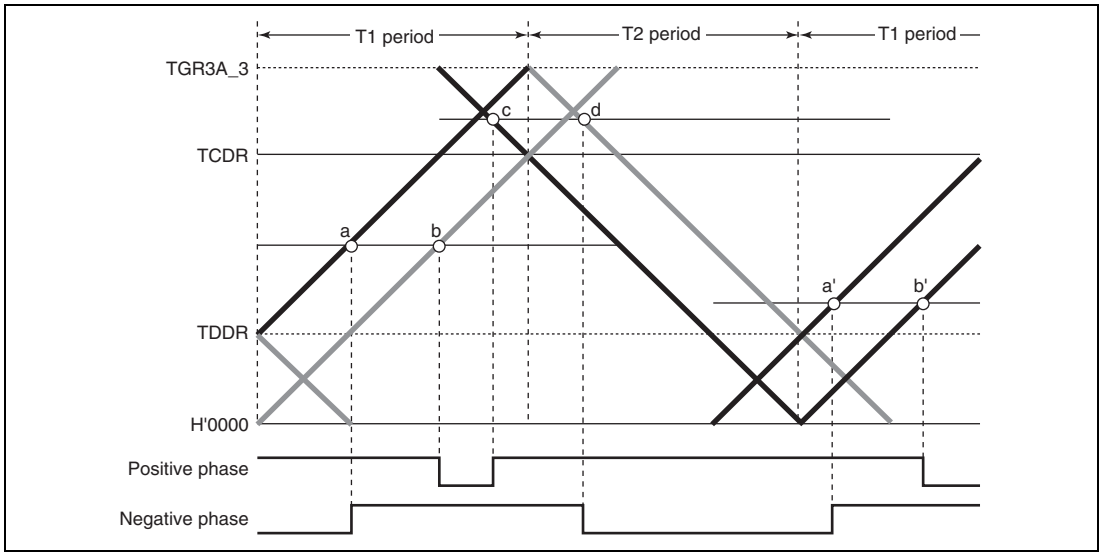


Figure 11.46 Example of Complementary PWM Mode Waveform Output (1)

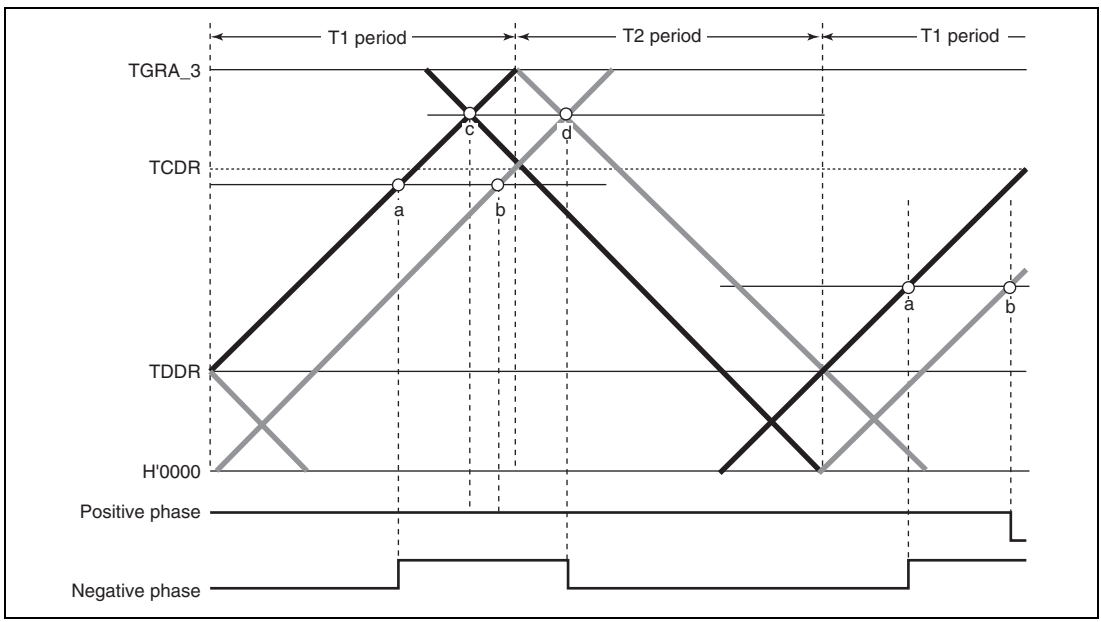


Figure 11.47 Example of Complementary PWM Mode Waveform Output (2)

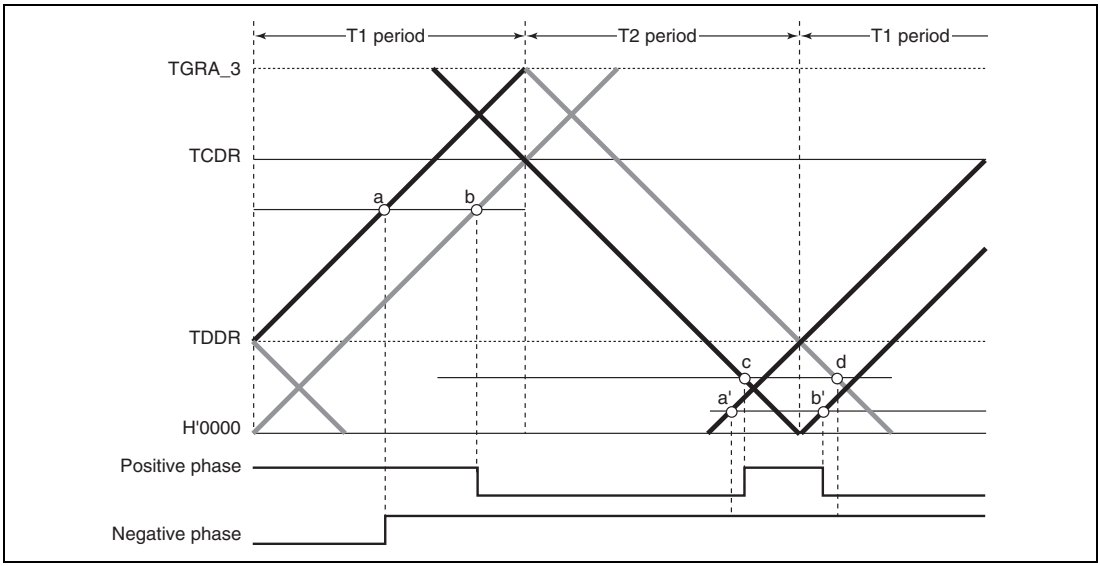


Figure 11.48 Example of Complementary PWM Mode Waveform Output (3)

(k) Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 11.49 to 11.53 show output examples.

100% duty output is performed when the compare register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the compare register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.

On and off compare-matches occur simultaneously, but if a turn-on compare-match and turn-off compare-match for the same phase occur simultaneously, both compare-matches are ignored and the waveform does not change.

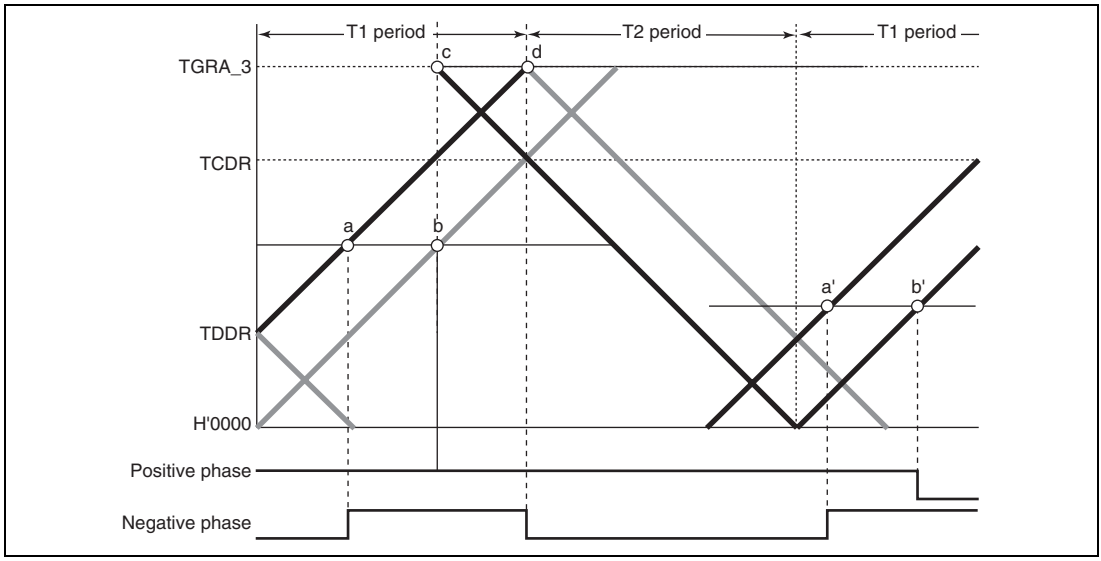


Figure 11.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

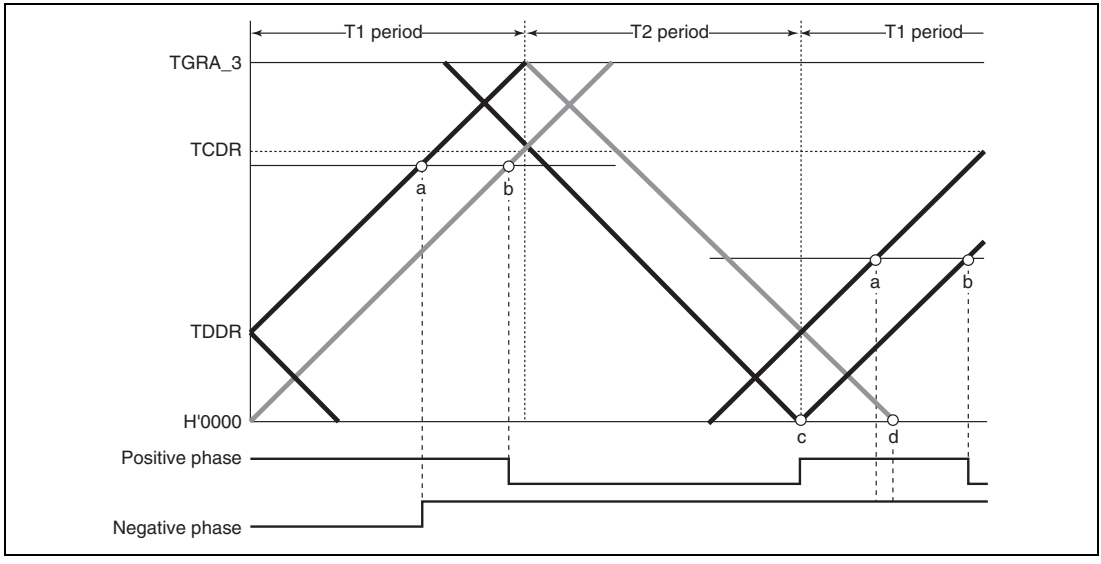


Figure 11.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

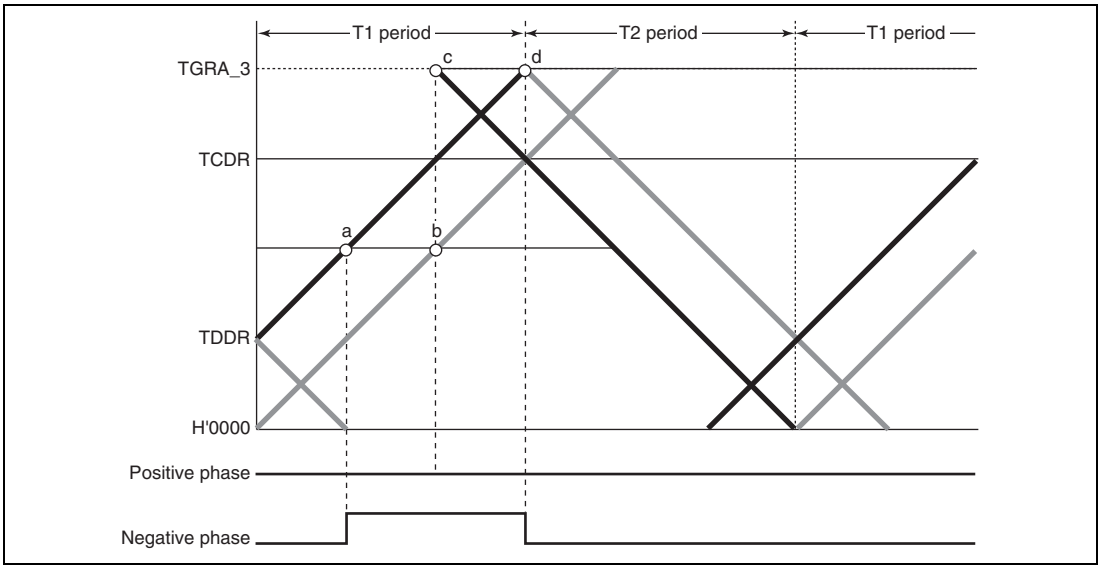


Figure 11.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

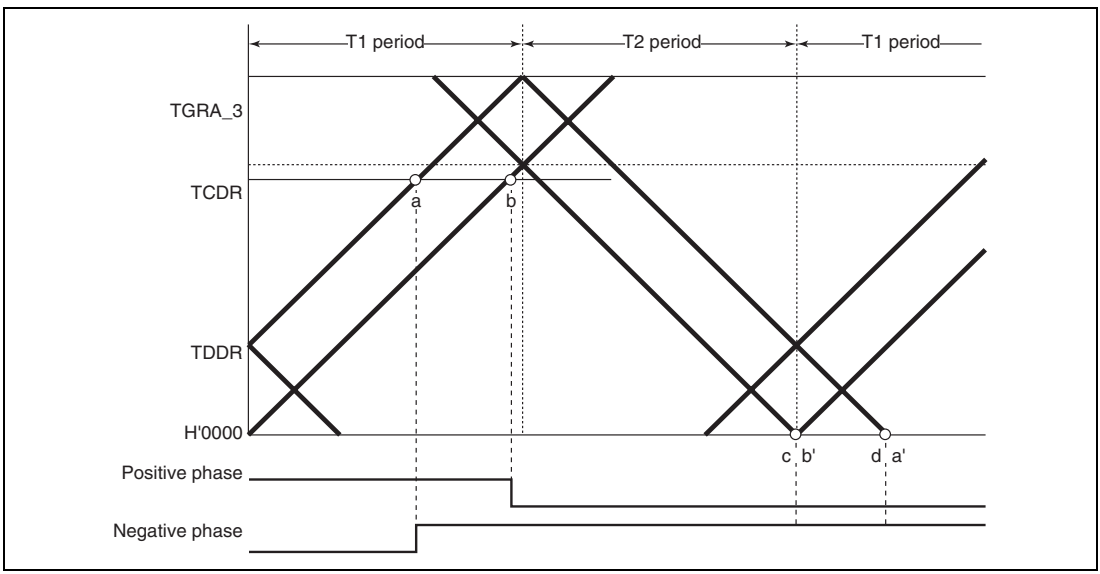


Figure 11.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

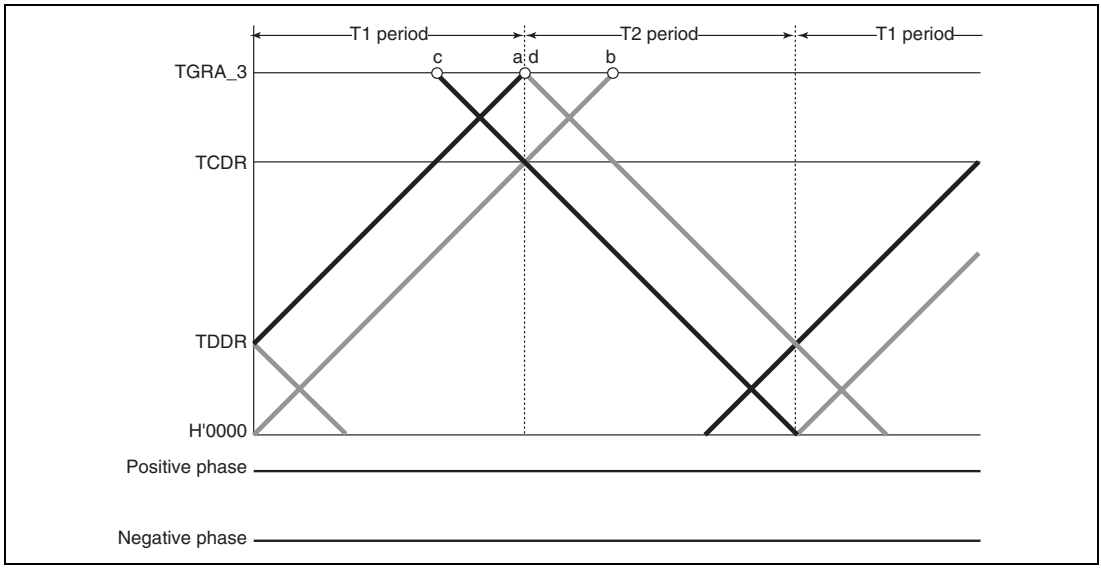


Figure 11.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output can be performed in synchronization with the PWM carrier cycle 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 11.54.

This output is toggled by a compare-match between TCNT_3 and TGRA_3 and a compare-match between TCNT4 and H'0000.

The output pin for this toggle output is the TIOC3A pin. The initial output is 1.

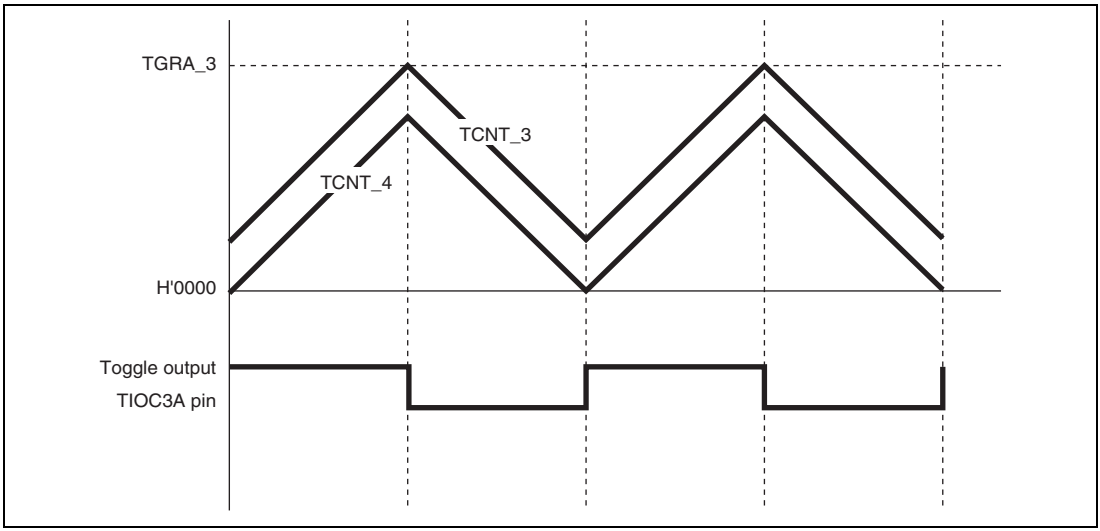


Figure 11.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, by setting a mode for synchronization with another channel by means of the timer synchronous register (TSYR), and selecting synchronous clearing with bits CCLR2 to CCLR0 in the timer control register (TCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by another channel.

Figure 11.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

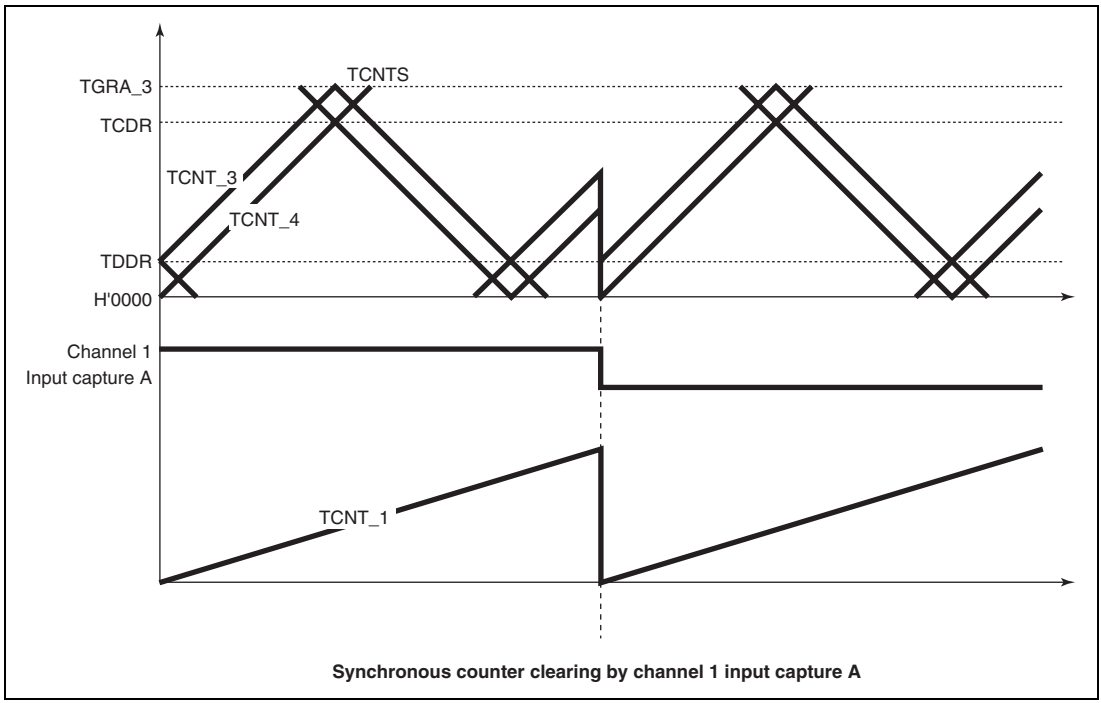


Figure 11.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in figure 11.56. 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 11.56) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both the MTU2 and MTU2S. In the MTU2, synchronous clearing generated in channels 0 to 2 in the MTU2 can cause counter clearing in complementary PWM mode; in the MTU2S, compare match or input capture flag setting in channels 0 to 2 in the MTU2 can cause counter clearing.

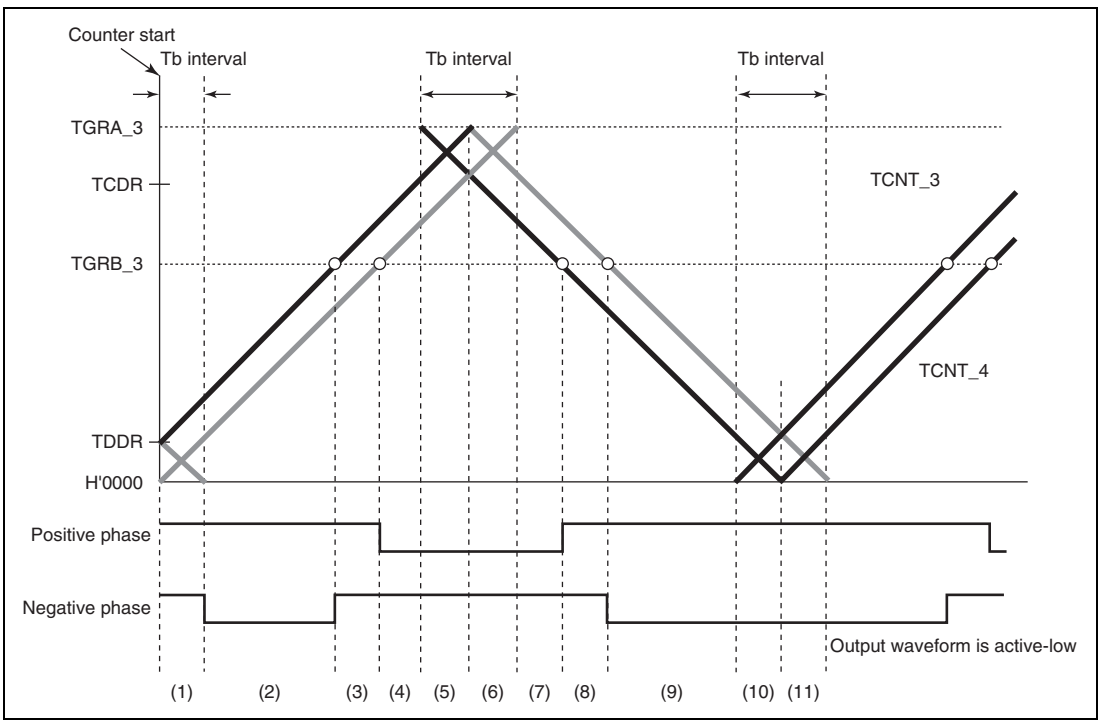


Figure 11.56 Timing for Synchronous Counter Clearing

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in figure 11.57.

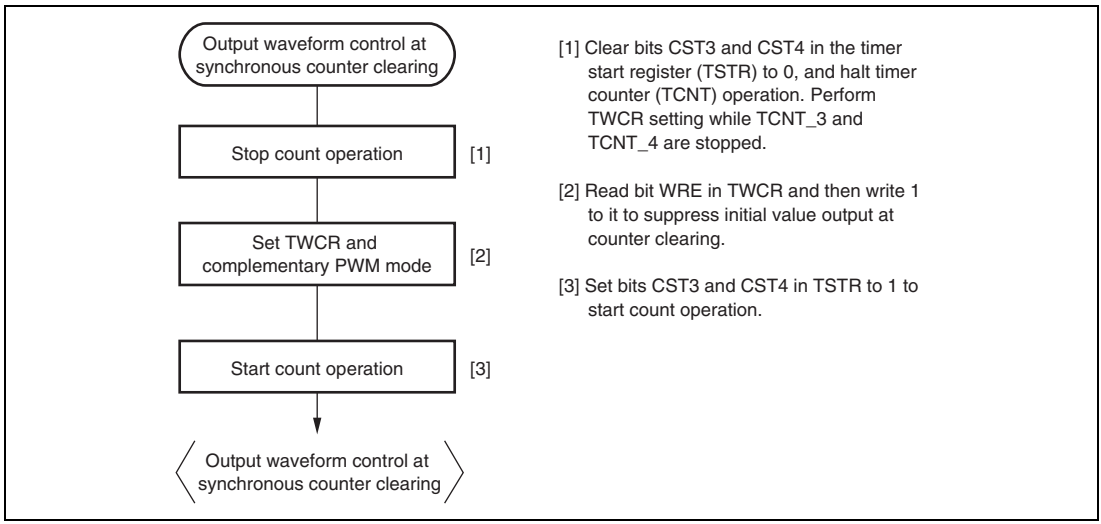


Figure 11.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 11.58 to 11.61 show examples of output waveform control in which the MTU2 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 figures 11.58 to 11.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 11.56, respectively.

  In the MTU2S, these examples are equivalent to the cases when the MTU2S operates in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is cleared to 0 and the WRE bit is set to 1 in TWCR.

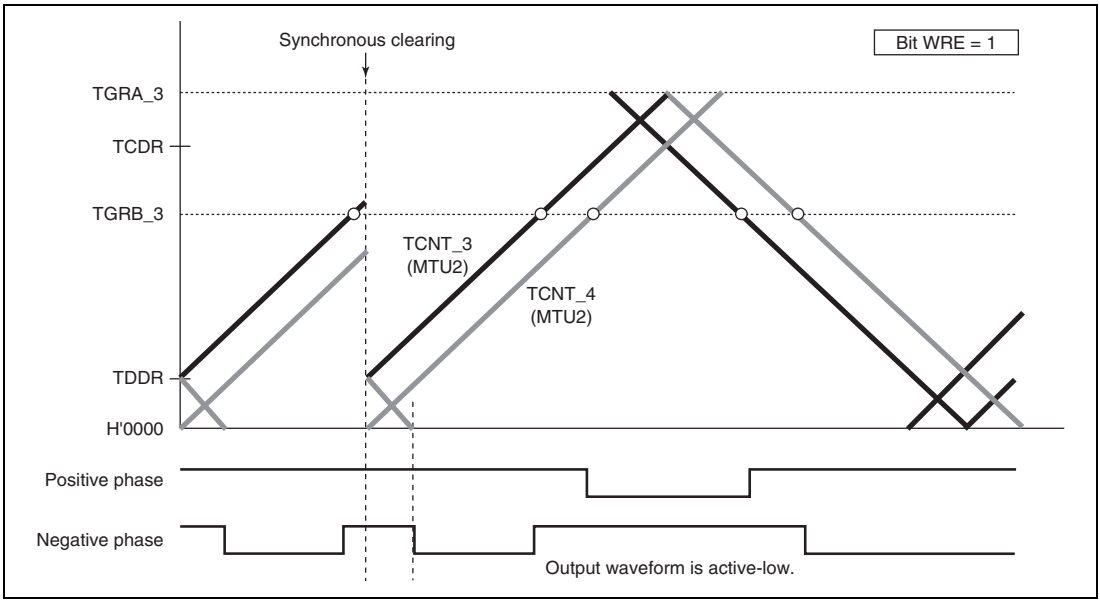


Figure 11.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 11.56; Bit WRE of TWCR in MTU2 is 1)

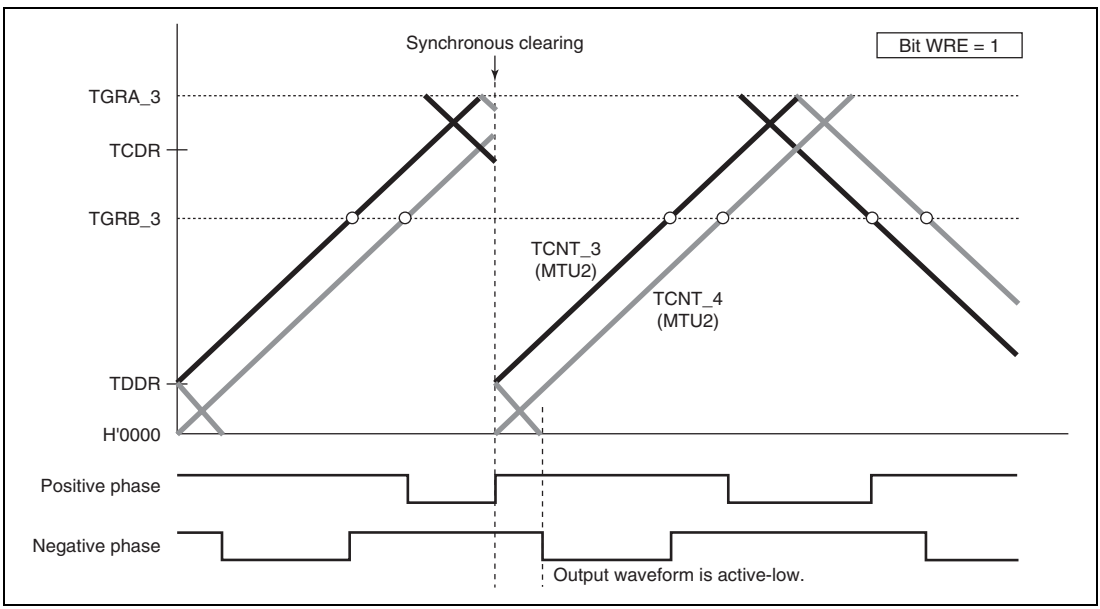


Figure 11.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 11.56; Bit WRE of TWCR in MTU2 is 1)

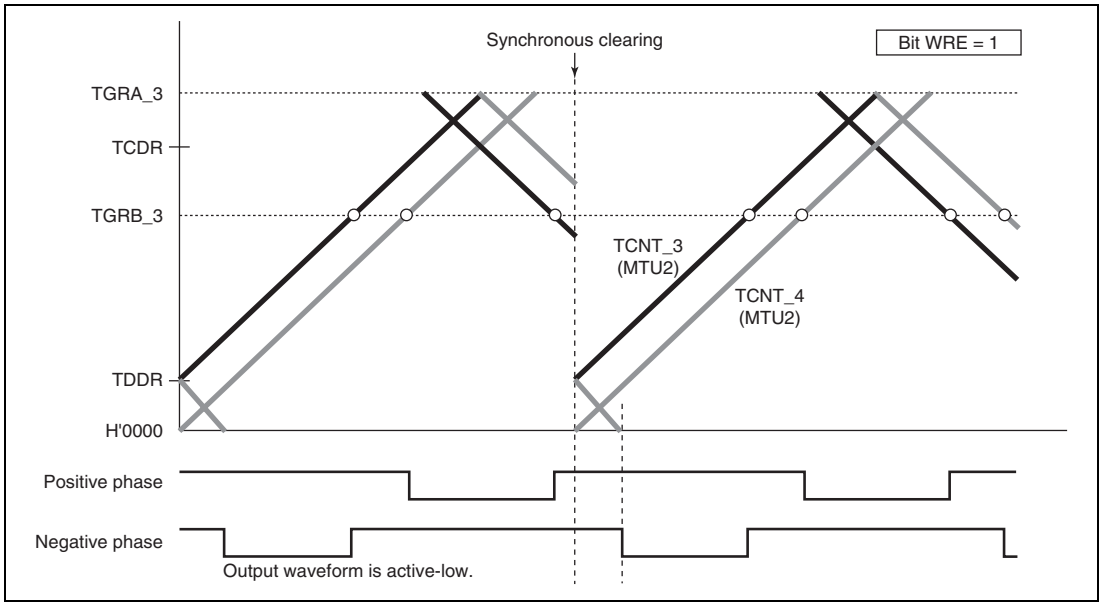


Figure 11.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 11.56; Bit WRE of TWCR is 1)

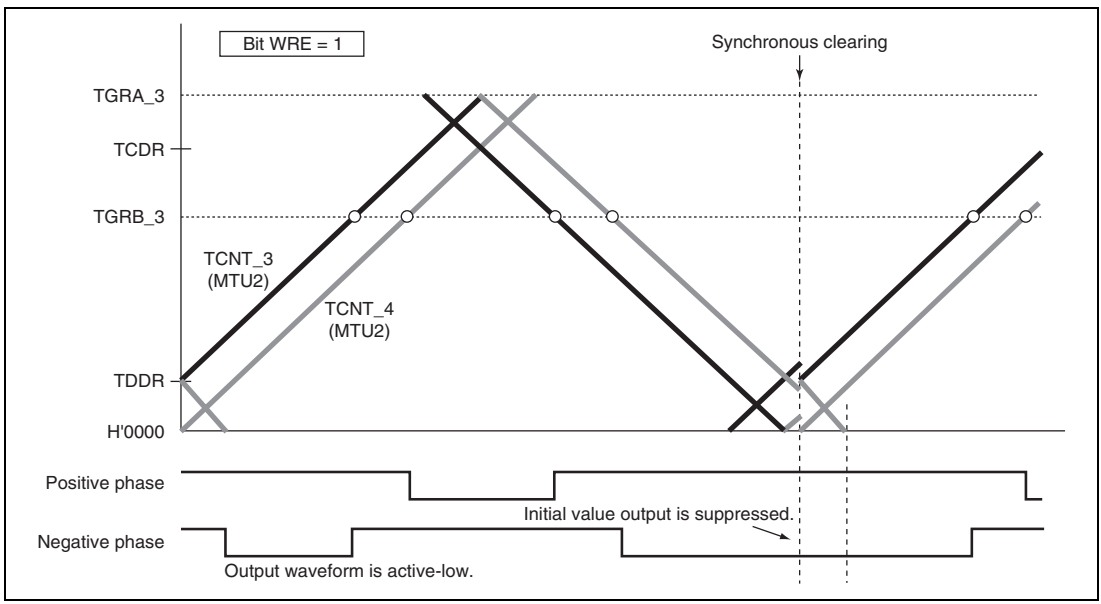


Figure 11.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 11.56; Bit WRE of TWCR is 1)

(o) Suppressing MTU2-MTU2S Synchronous Counter Clearing

In the MTU2S, setting the SCC bit in TWCR to 1 suppresses synchronous counter clearing caused by the MTU2.

Synchronous counter clearing is suppressed only within the interval shown in figure 11.62. When using this function, the MTU2S should be set to complementary PWM mode.

For details of synchronous clearing caused by the MTU2, refer to the description about MTU2S counter clearing caused by MTU2 flag setting source (MTU2-MTU2S synchronous counter clearing) in section 11.4.10, MTU2-MTU2S Synchronous Operation.

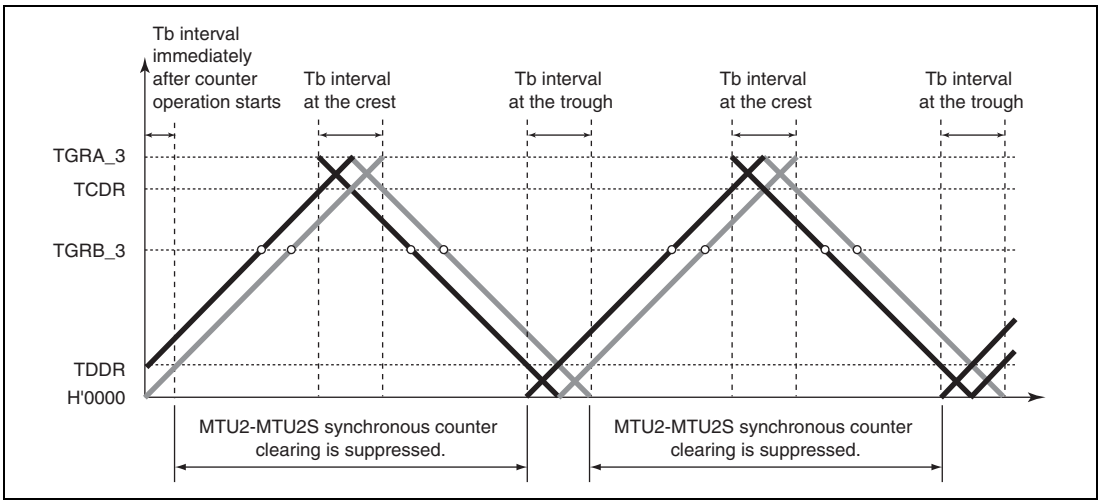


Figure 11.62 MTU2-MTU2S Synchronous Clearing-Suppressed Interval Specified by SCC Bit in TWCR

- Example of Procedure for Suppressing MTU2-MTU2S Synchronous Counter Clearing

  An example of the procedure for suppressing MTU2-MTU2S synchronous counter clearing is shown in figure 11.63.

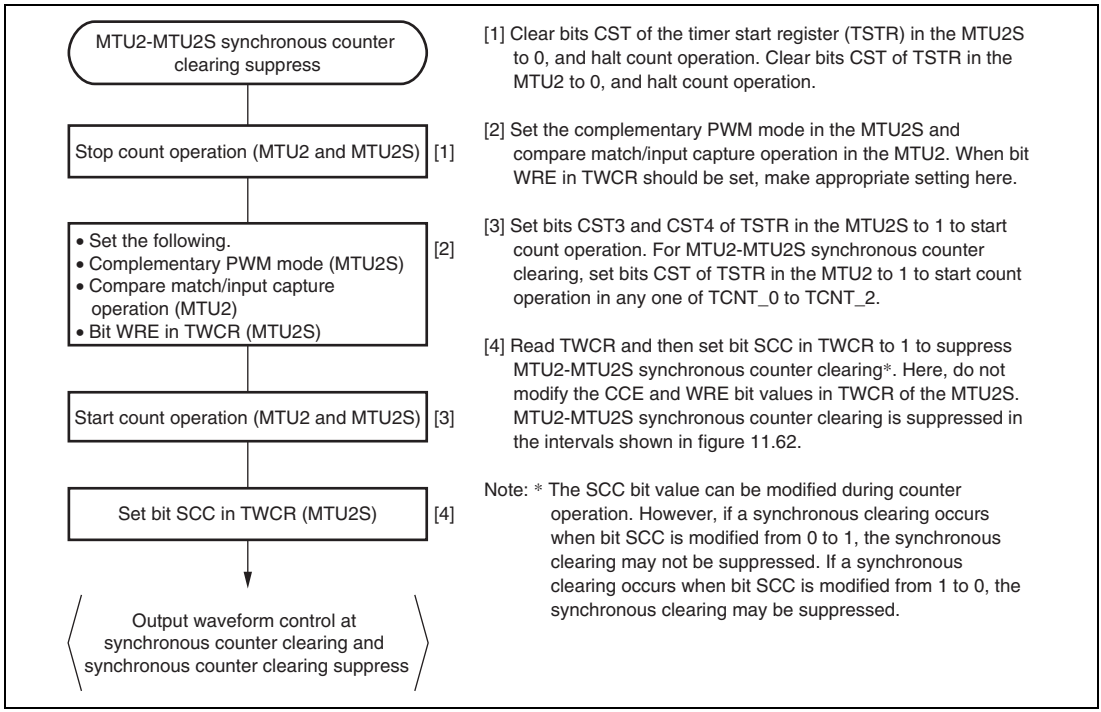


Figure 11.63 Example of Procedure for Suppressing MTU2-MTU2S Synchronous Counter Clearing

- Examples of Suppression of MTU2-MTU2S Synchronous Counter Clearing

  Figures 11.64 to 11.67 show examples of operation in which the MTU2S operates in complementary PWM mode and MTU2-MTU2S synchronous counter clearing is suppressed by setting the SCC bit in TWCR in the MTU2S to 1. In the examples shown in figures 11.64 to 11.67, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 11.56, respectively.

  In these examples, the WRE bit in TWCR of the MTU2S is set to 1.

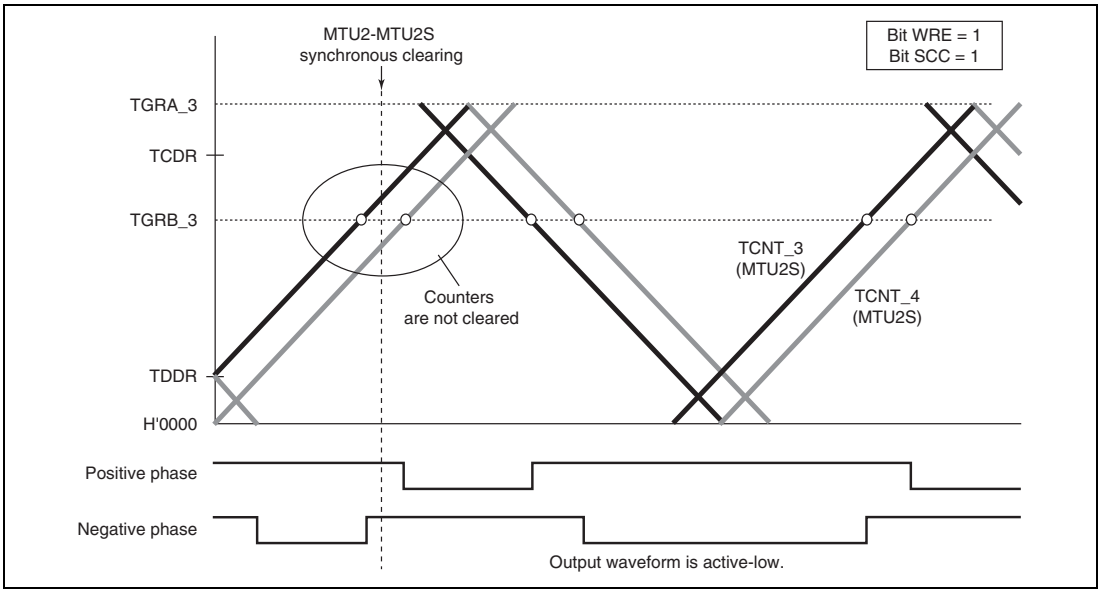


Figure 11.64 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

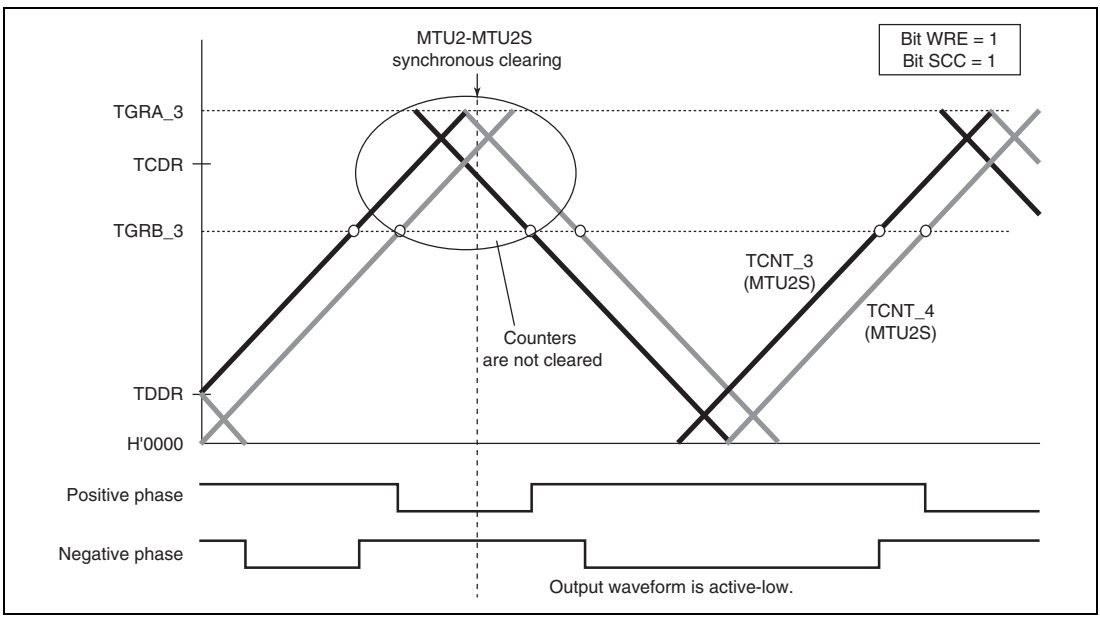


Figure 11.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

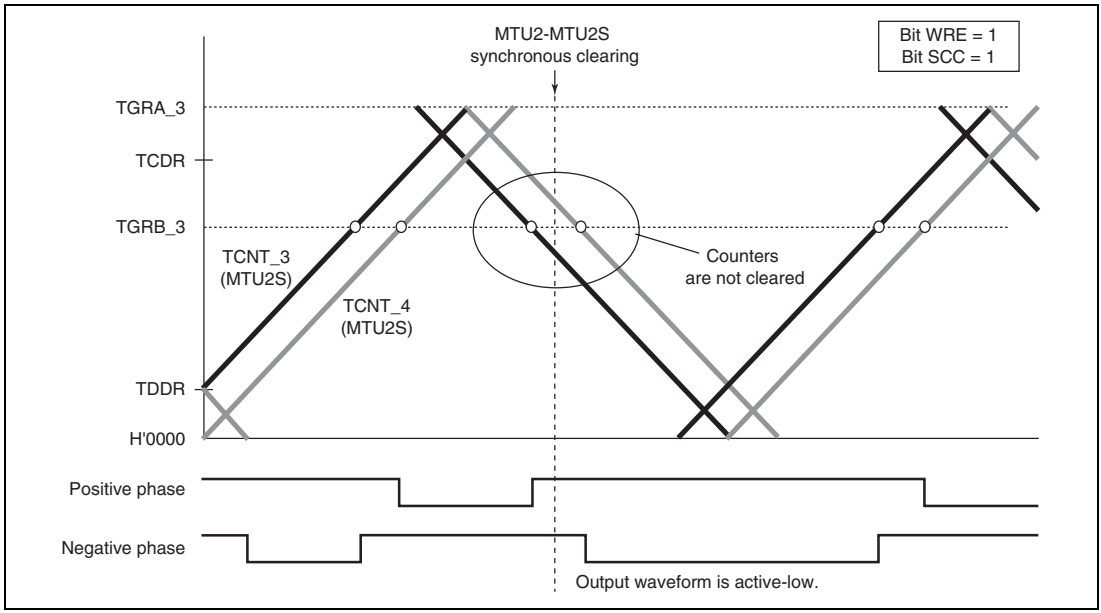


Figure 11.66 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

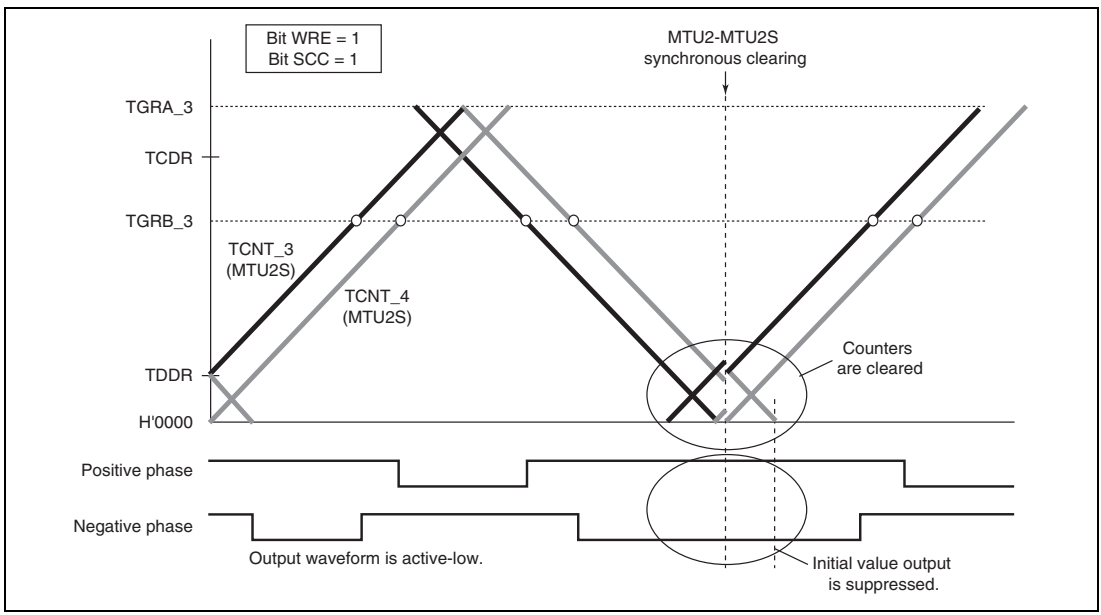


Figure 11.67 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

(p) Counter Clearing by TGRA_3 Compare Match

In complementary PWM mode, by setting the CCE bit in the timer waveform control register (TWCR), it is possible to have TCNT_3, TCNT_4, and TCNTS cleared by TGRA_3 compare match.

Figure 11.68 illustrates an operation example.

Notes: 1. Use this function only in complementary PWM mode 1 (transfer at crest)
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, CE0B, CE0C, CE0D, CE1A, CE1B, CE1C, and CE1D bits in the timer synchronous clear register (TSYCR) to 1).
3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.

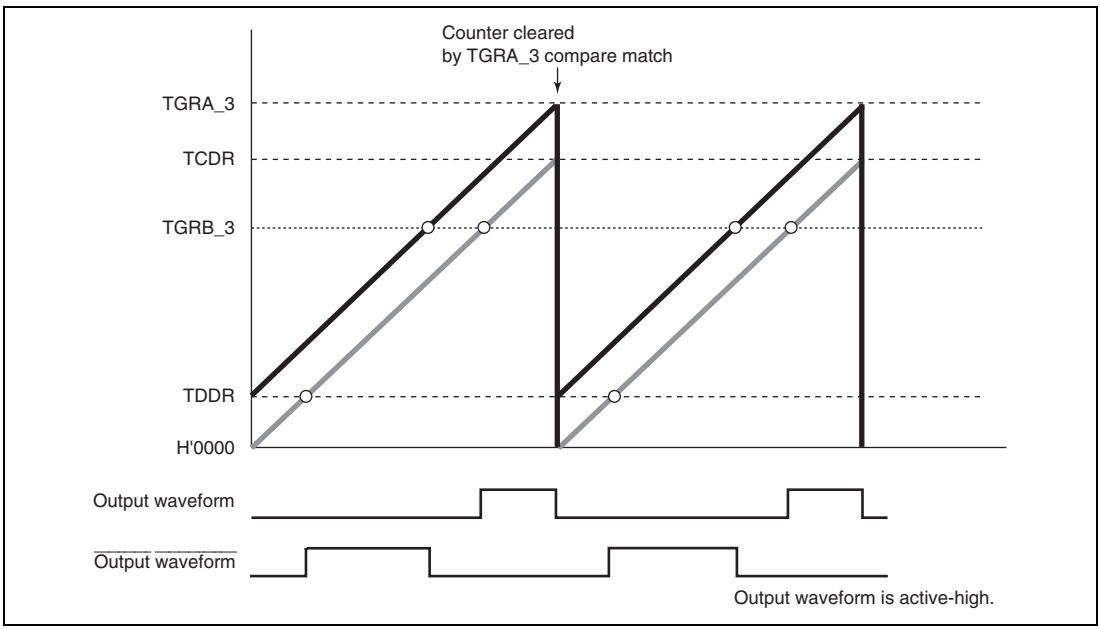


Figure 11.68 Example of Counter Clearing Operation by TGRA_3 Compare Match

(q) Example of AC Synchronous Motor (Brushless DC Motor) Drive Waveform Output

In complementary PWM mode, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figures 11.69 to 11.72 show examples of brushless DC motor drive waveforms created using TGCR.

When output phase switching for a 3-phase brushless DC motor is performed 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 polarity position are input to channel 0 timer input pins TIOC0A, TIOC0B, and TIOC0C (set with PFC). When an edge is detected at pin TIOC0A, TIOC0B, or TIOC0C, 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 drive waveforms are output from the complementary PWM mode 6-phase output pins. With this 6-phase output, in the case of on output, it is possible to use complementary PWM mode output and perform chopping output by setting the N bit or P bit to 1. When the N bit or P bit is 0, level output is selected.

The 6-phase output active level (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.

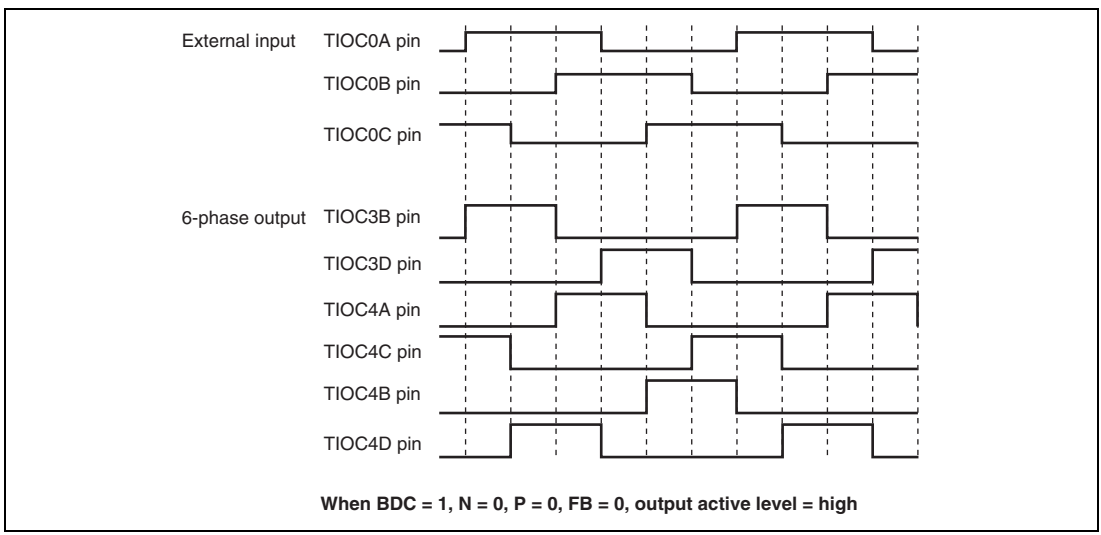


Figure 11.69 Example of Output Phase Switching by External Input (1)

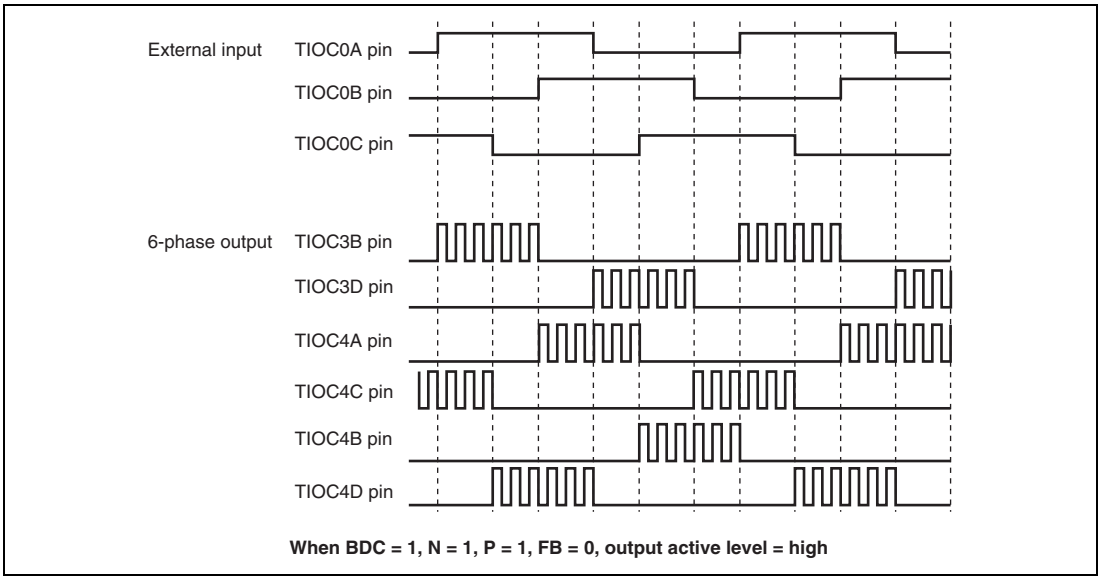


Figure 11.70 Example of Output Phase Switching by External Input (2)

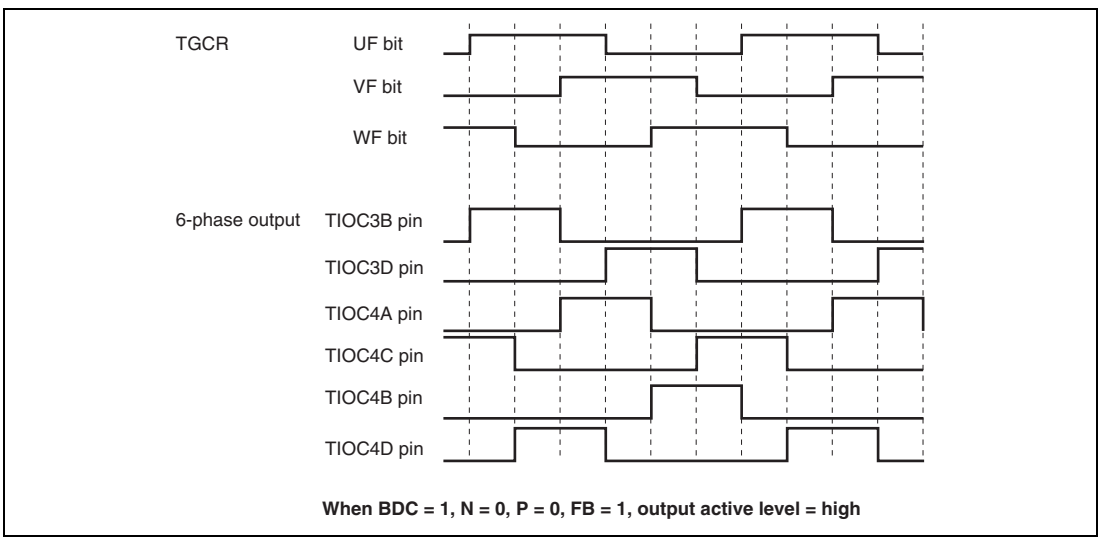


Figure 11.71 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

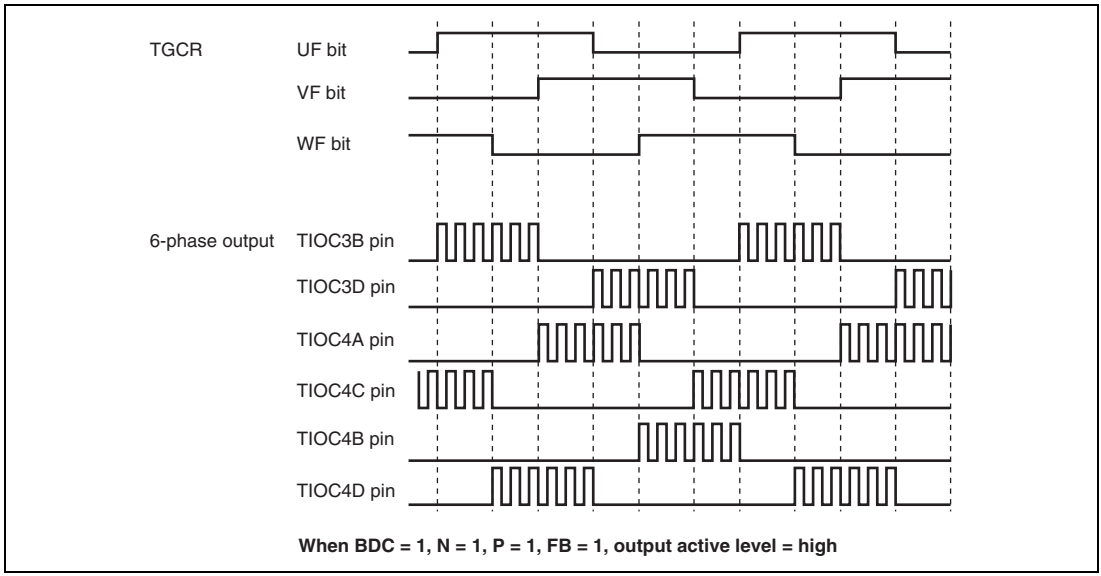


Figure 11.72 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(r) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using a TGRA_3 compare-match, TCNT_4 underflow (trough), or compare-match on a channel other than channels 3 and 4.

When start requests using a TGRA_3 compare-match are specified, A/D conversion can be started at the crest of the TCNT_3 count.

A/D converter start requests can be set by setting the TTGE bit to 1 in the timer interrupt enable register (TIER). To issue an A/D converter start request at a TCNT_4 underflow (trough), set the TTGE2 bit in TIER_4 to 1.

(3) Interrupt Skipping in Complementary PWM Mode

Interrupts TGIA_3 (at the crest) and TCIV_4 (at the trough) in channels 3 and 4 can be skipped up to seven times by making settings in the timer interrupt skipping set register (TITCR).

Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR). For the linkage with the A/D converter start request delaying function, refer to section 11.4.9, A/D Converter Start Request Delaying Function.

The setting of the timer interrupt skipping setting register (TITCR) must be done while the TGIA_3 and TCIV_4 interrupt requests are disabled by the settings of TIER_3 and TIER_4 along with under the conditions in which TGFA_3 and TCFV_4 flag settings by compare match never occur. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 11.73 shows an example of the interrupt skipping operation setting procedure. Figure 11.74 shows the periods during which interrupt skipping count can be changed.

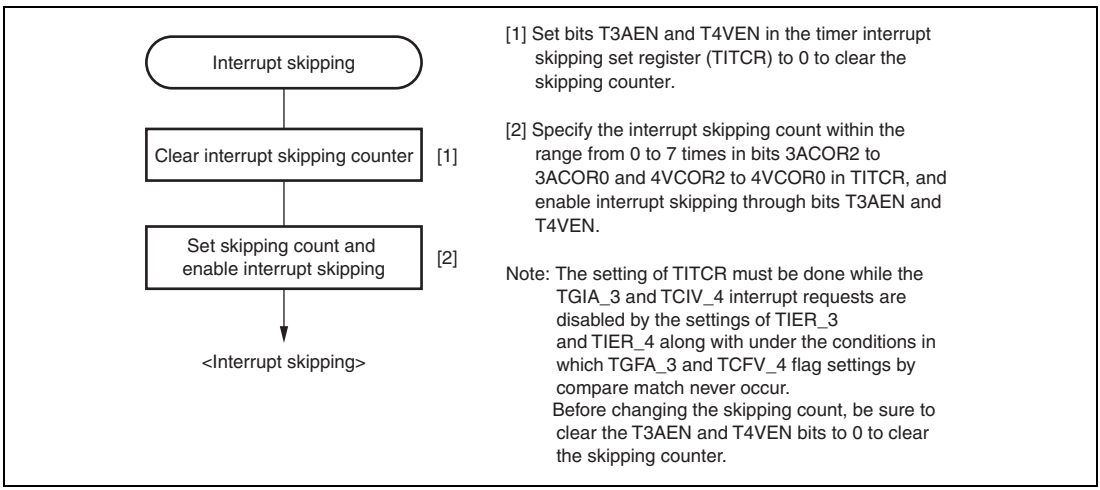


Figure 11.73 Example of Interrupt Skipping Operation Setting Procedure

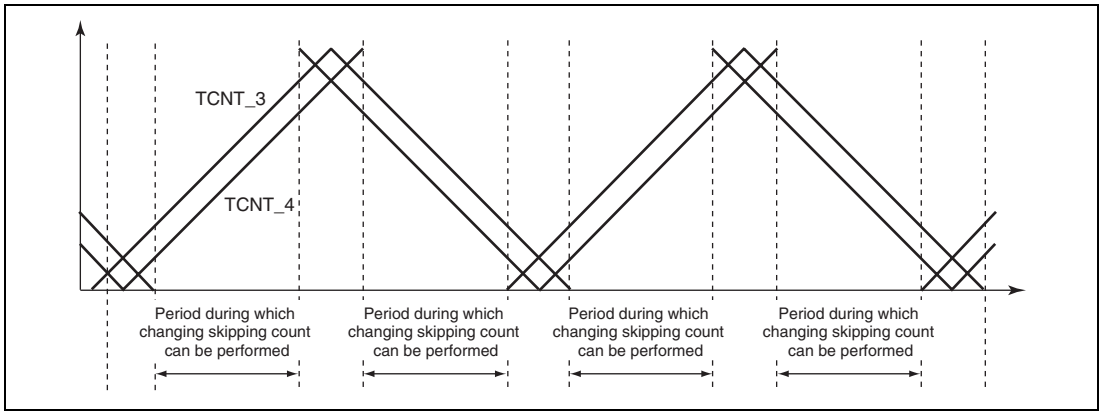


Figure 11.74 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 11.75 shows an example of TGIA_3 interrupt skipping in which the interrupt skipping count is set to three by the 3ACOR bit and the T3AEN bit is set to 1 in the timer interrupt skipping set register (TITCR).

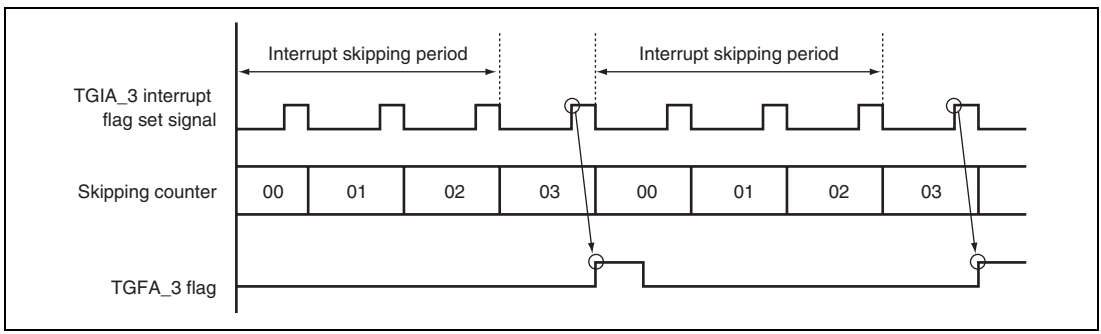


Figure 11.75 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

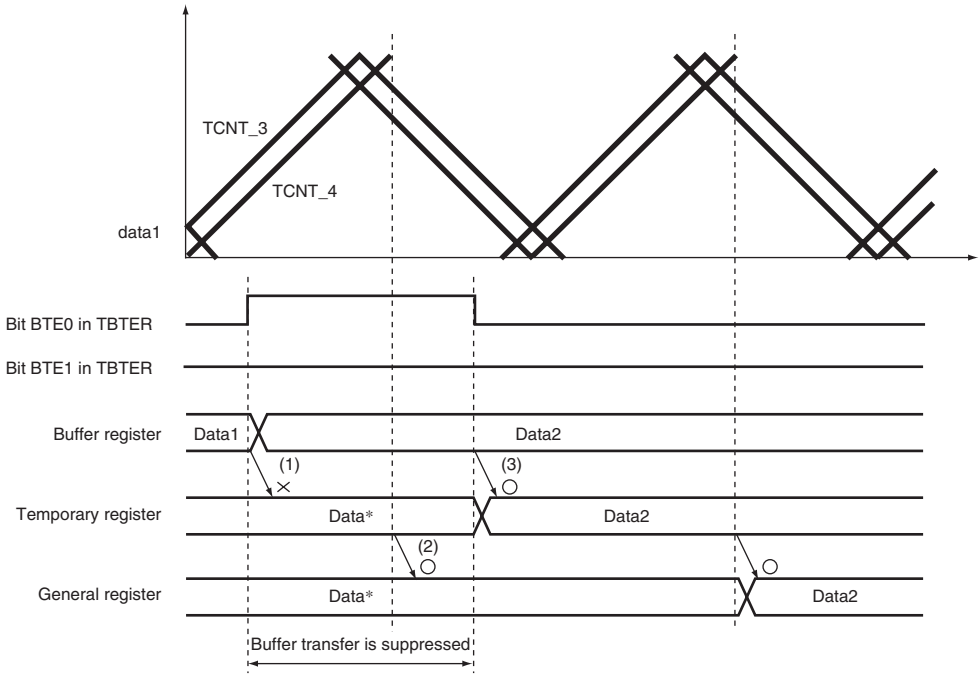
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

Figure 11.76 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 11.77 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE1 = 1 and BET0 = 0). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. Depending on the timing of interrupt generation and writing to the buffer register, the timing of transfer from the buffer register to the temporary register and from the temporary register to the general register is one of two types.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 11.78 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 (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

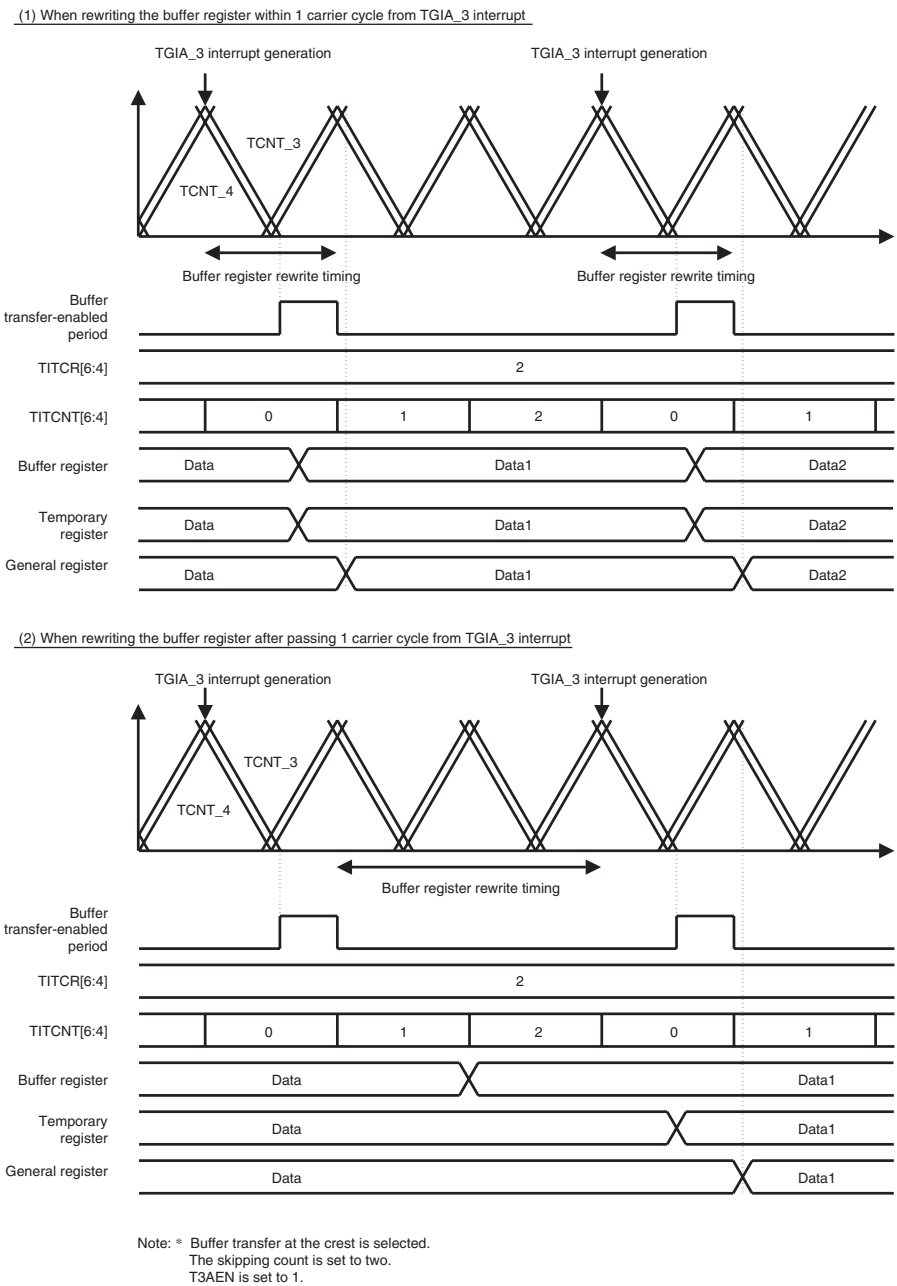


[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note: * When buffer transfer at the crest is selected.

Figure 11.76 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



Note: * Buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN is set to 1.

Figure 11.77 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)

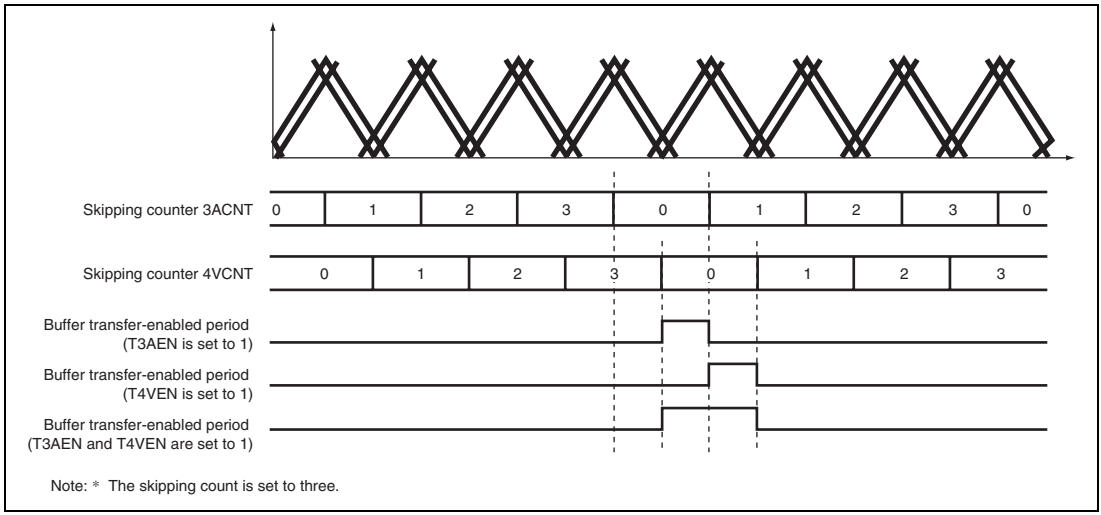


Figure 11.78 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Function

Complementary PWM mode output has the following protection functions.

(a) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be rewritten 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 (21 in total) of the registers in channels 3 and 4 shown in the following:

- TCR_3 and TCR_4, TMDR_3 and TMDR_4, TIORH_3 and TIORH_4, TIORL_3 and TIORL_4, TIER_3 and TIER_4, TCNT_3 and TCNT_4, TGRA_3 and TGRA_4, TGRB_3 and TGRB_4, TOER, TOCR, TGCR, TCDR, and TDDR.

This function enables miswriting due to CPU runaway to be prevented by disabling CPU access to the mode registers, control registers, and counters. When the applicable registers are read in the access-disabled state, undefined values are returned. Writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

The 6-phase PWM output pins can be set automatically to the high-impedance state by inputting specified external signals. There are four external signal input pins.

See section 13, Port Output Enable 2 (POE2), for details.

(c) Halting of PWM Output by Oscillation Stop

The 6-phase PWM output pins can detect the clock stop and set the output pin automatically to the high-impedance state. However, the pin state is not guaranteed when the clock starts oscillation again.

See section 4.7, Oscillation Stop Detection, for details.

11.4.9 A/D Converter Start Request Delaying Function

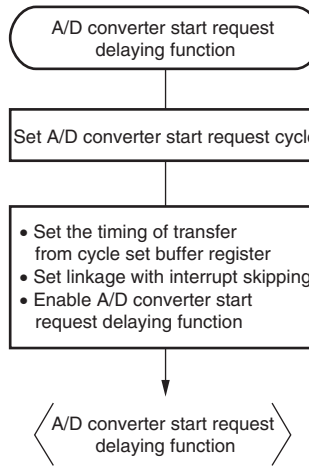
A/D converter start requests can be issued in channel 4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4), and timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4).

The A/D converter start request delaying function compares TCNT_4 with TADCORA_4 or TADCORB_4, and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN).

A/D converter start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in TADCR.

- Example of Procedure for Specifying A/D Converter Start Request Delaying Function

  Figure 11.79 shows an example of procedure for specifying the A/D converter start request delaying function.



[1] Set the cycle in the timer A/D converter start request cycle buffer register (TADCOBRA_4 or TADCOBRB_4) and timer A/D converter start request cycle register (TADCORA_4 or TADCORB_4). (The same initial value must be specified in the cycle buffer register and cycle register.)

[2] Use bits BF1 and BF2 in the timer A/D converter start request control register (TADCR) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits TU4AE, DT4AE, UT4BE, and DT4BE to enable A/D conversion start requests (TRG4AN or TRG4BN).

Notes: 1. Perform TADCR setting while TCNT_4 is stopped.
2. Do not set BF1 to 1 when complementary PWM mode is not selected.
3. Do not set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, or DT4BE to 1 when complementary PWM mode is not selected.

Figure 11.79 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 11.80 shows a basic example of A/D converter request signal (TRG4AN) operation when the trough of TCNT_4 is specified for the buffer transfer timing and an A/D converter start request signal is output during TCNT_4 down-counting.

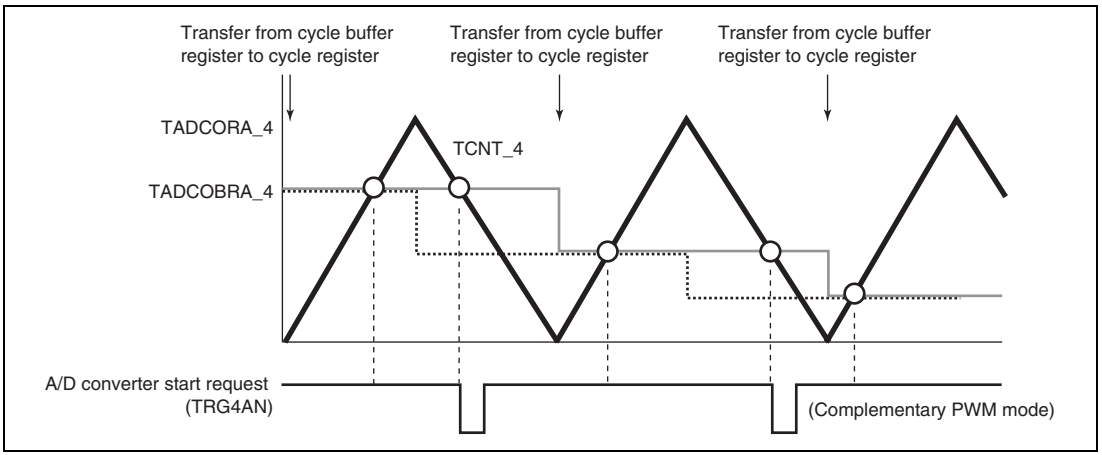


Figure 11.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- Buffer Transfer

  The data in the timer A/D converter start request cycle set registers (TADCORA_4 and TADCORB_4) is updated by writing data to the timer A/D converter start request cycle set buffer registers (TADCOBRA_4 and TADCOBRB_4). 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 (TADCR_4).

- A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

  A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR).

  Figure 11.81 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

  Figure 11.82 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during TCNT_4 up-counting and A/D converter start requests are linked with interrupt skipping.

Note: This function must 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 (3ACOR and 4VCOR) 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).

Furthermore, when this function is to be used, set TADCORA_4 and TADCORB_4 to a value between H'0002 and the TCDR setting minus two.

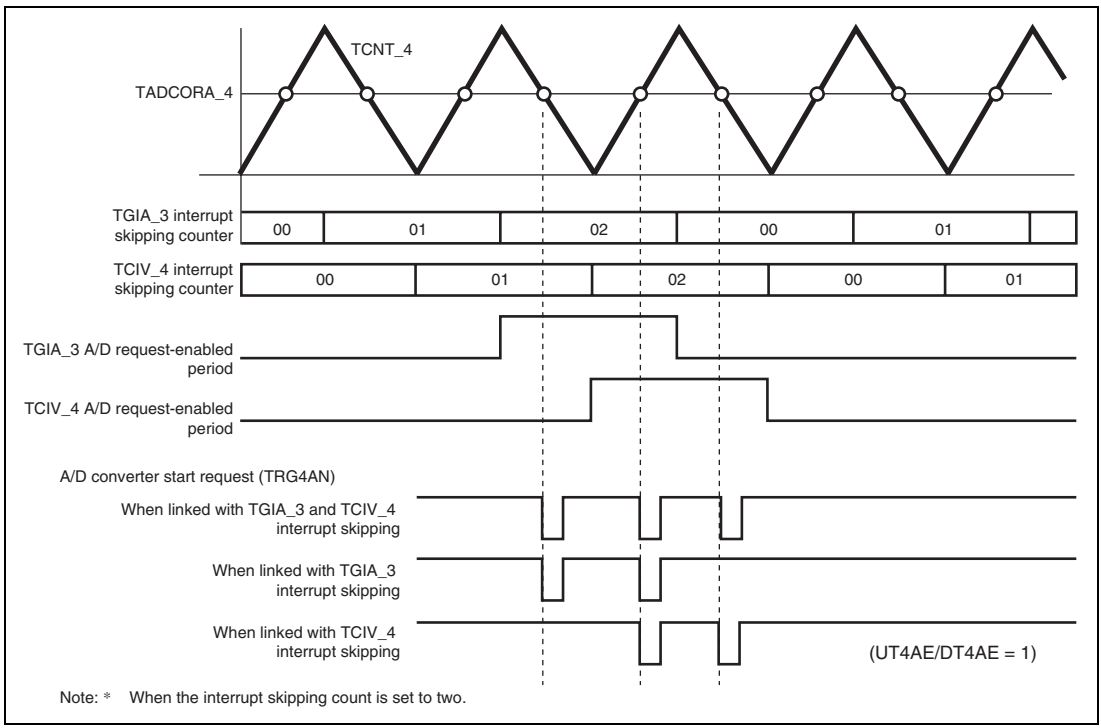


Figure 11.81 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

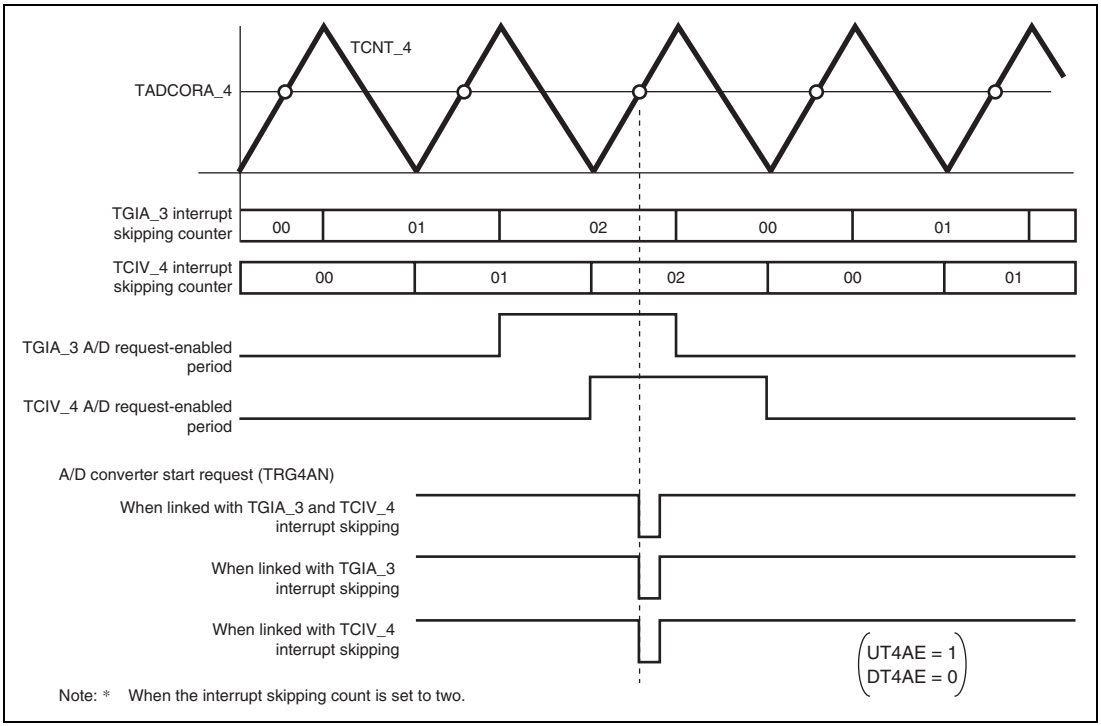


Figure 11.82 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

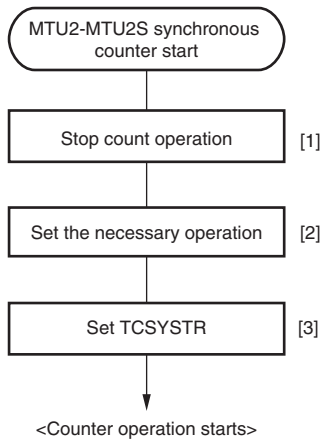
11.4.10 MTU2-MTU2S Synchronous Operation

(1) MTU2-MTU2S Synchronous Counter Start

The counters in the MTU2 and MTU2S which operate at different clock systems can be started synchronously by making the TCSYSTR settings in the MTU2.

(a) Example of MTU2-MTU2S Synchronous Counter Start Setting Procedure

Figure 11.83 shows an example of synchronous counter start setting procedure.



[1] Use TSTR registers in the MTU2 and MTU2S and halt the counters used for synchronous start operation.

[2] Specify necessary operation with appropriate registers such as TCR and TMDR.

[3] In TCSYSTR in the MTU2, set the bits corresponding to the counters to be started synchronously to 1. The TSTRs are automatically set appropriately and the counters start synchronously.

Notes: 1. Even if a bit in TCSYSTR corresponding to an operating counter is cleared to 0, the counter will not stop. To stop the counter, clear the corresponding bit in TSTR to 0 directly.

2. To start channels 3 and 4 in reset-synchronized PWM mode or complementary PWM mode, make appropriate settings in TCYSTR according to the TSTR setting for the respective mode. For details, refer to section 11.4.7, Reset-Synchronized PWM Mode, and section 11.4.8, Complementary PWM Mode.

Figure 11.83 Example of Synchronous Counter Start Setting Procedure

(b) Examples of Synchronous Counter Start Operation

Figures 11.84 (1) to (4) show examples of synchronous counter start operation when the clock frequency ratios between the MTU2 and MTU2S are 1:1, 1:2, 1:3, and 1:4, respectively. In these examples, the count clock is set to Pϕ/1.

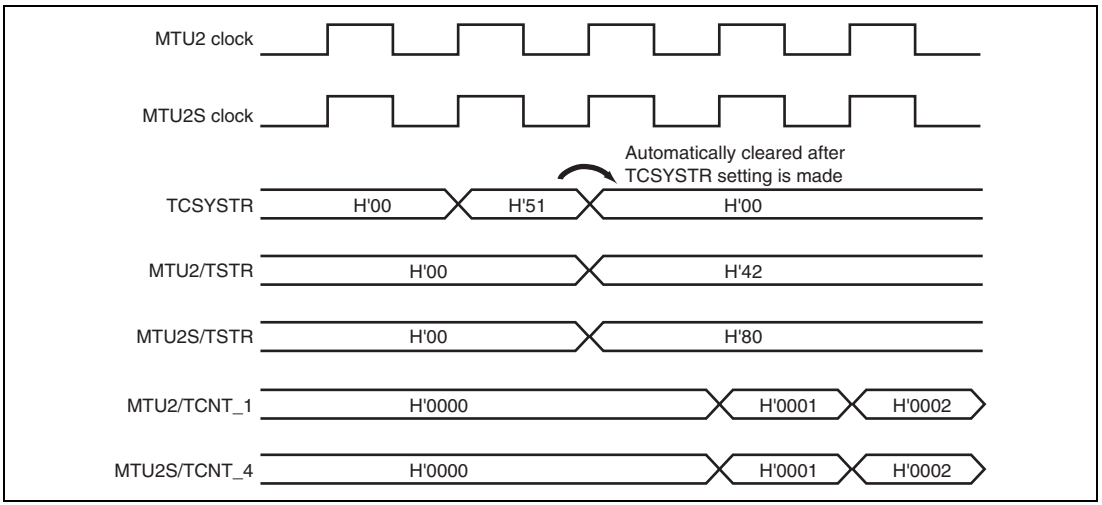


Figure 11.84 (1) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:1)

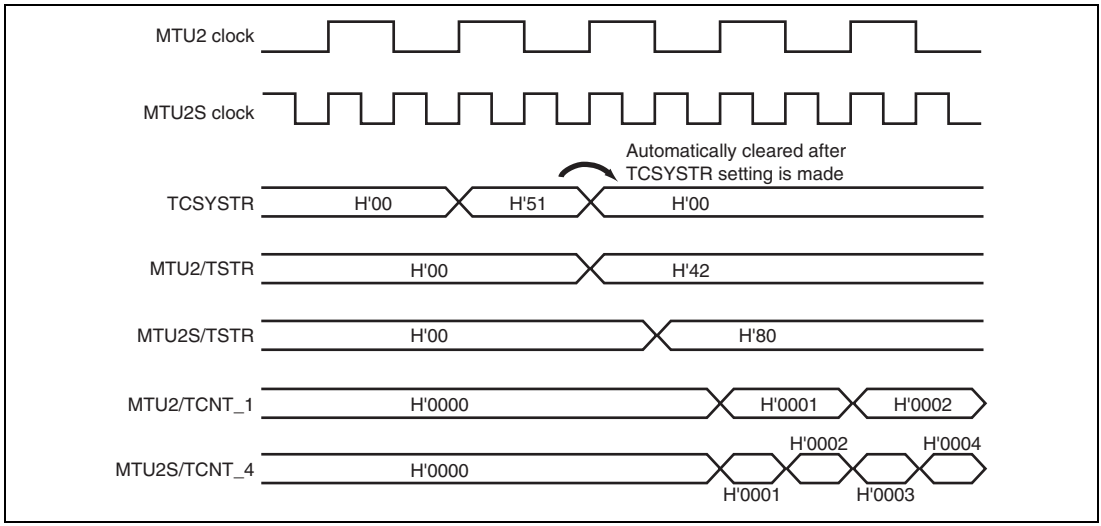


Figure 11.84 (2) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:2)

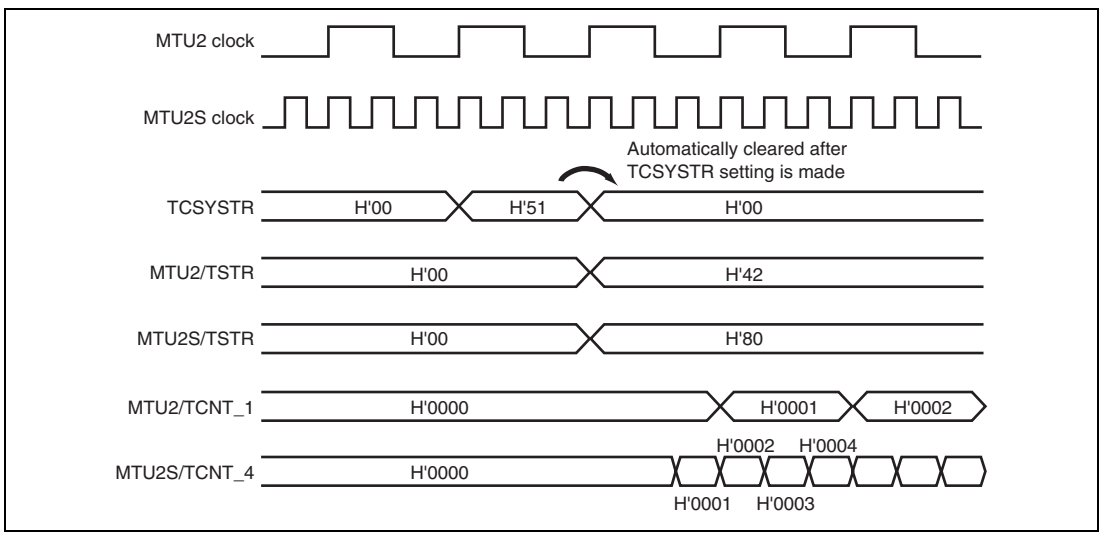


Figure 11.84 (3) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:3)

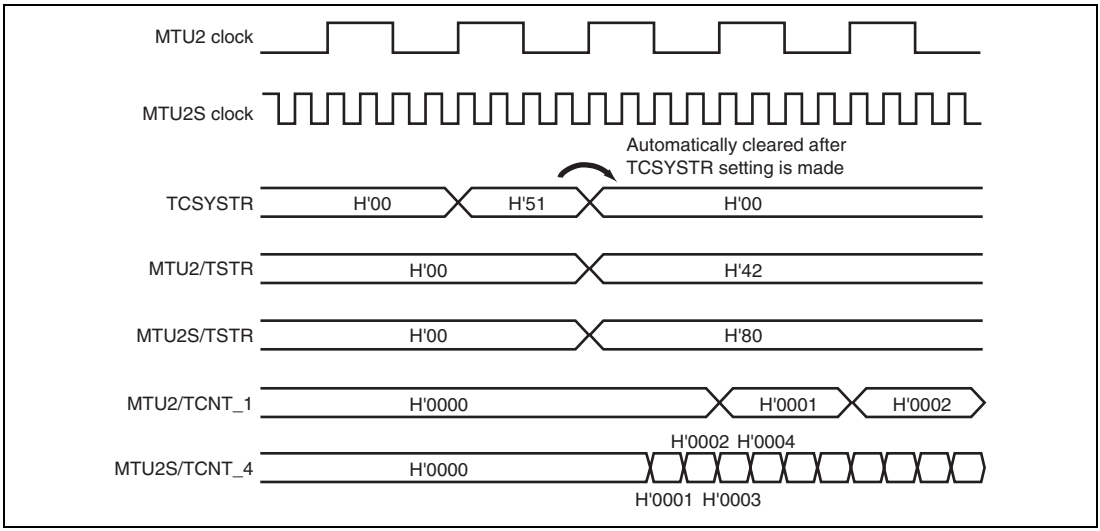


Figure 11.84 (4) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:4)

(2) MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (MTU2-MTU2S Synchronous Counter Clearing)

The MTU2S counters can be cleared by sources for setting the flags in TSR_0 to TSR_2 in the MTU2 through the TSYCR_S settings in the MTU2S.

(a) Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

Figure 11.85 shows an example of procedure for specifying MTU2S counter clearing by MTU2 flag setting source.

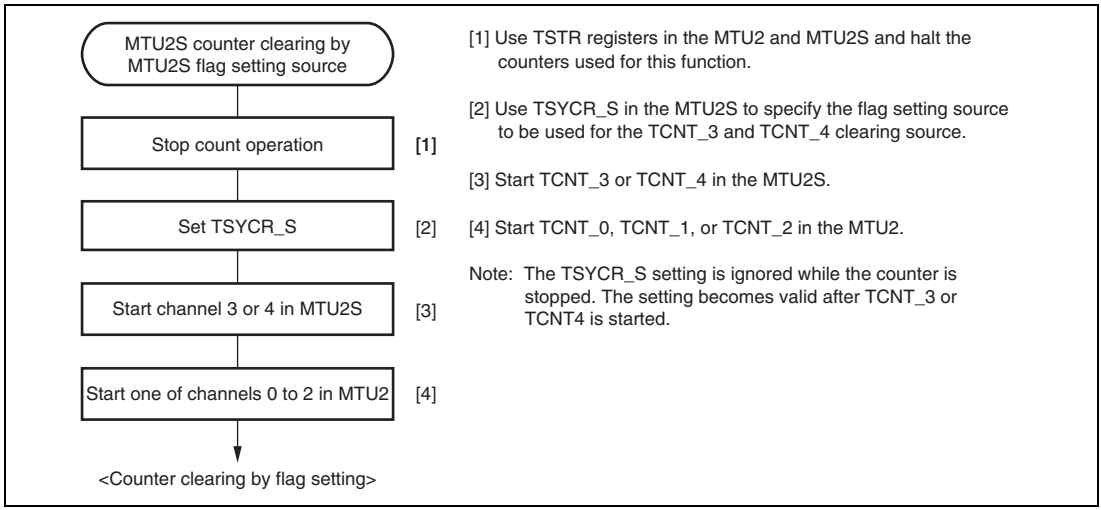


Figure 11.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

(b) Examples of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source

Figures 11.86 (1) and 11.86 (2) show examples of MTS2S counter clearing caused by MTU2 flag setting source.

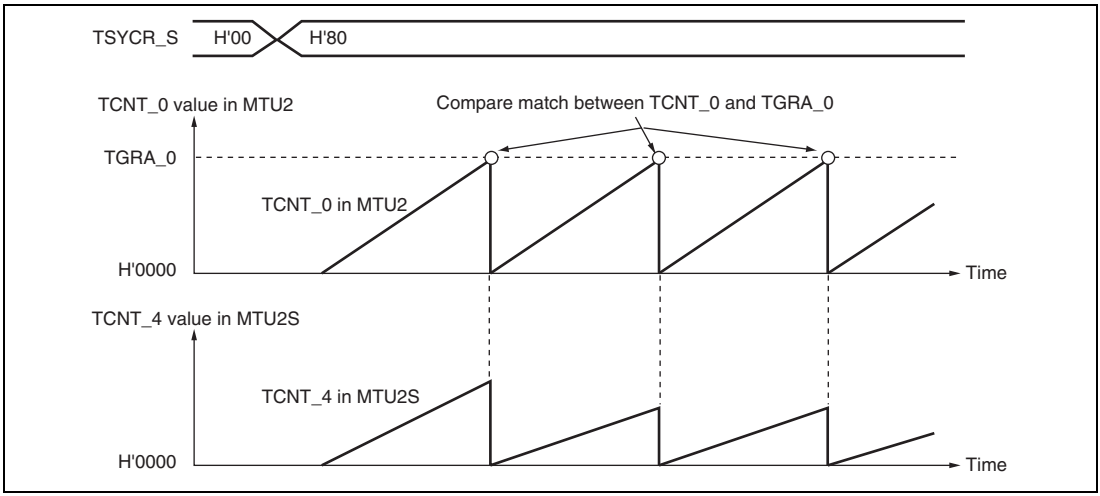


Figure 11.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)

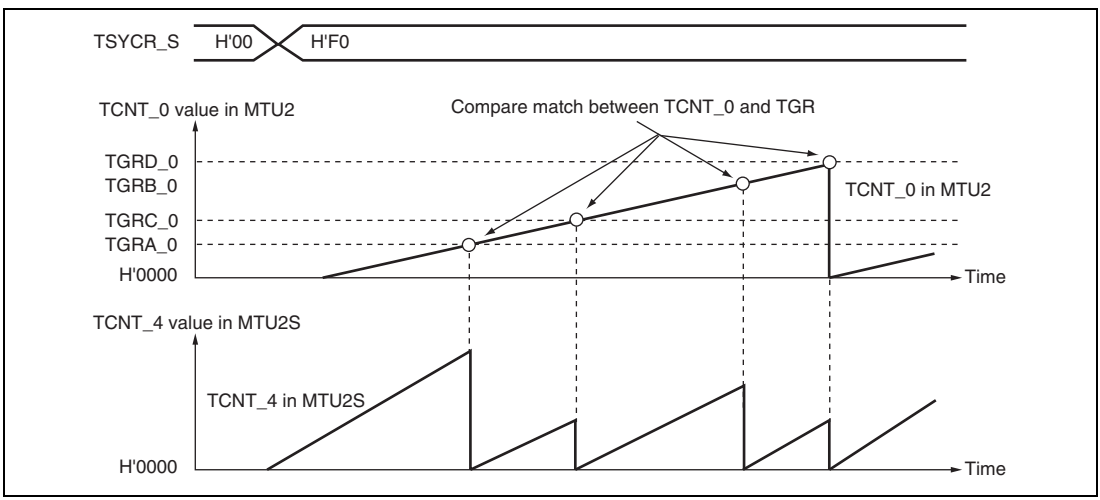


Figure 11.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)

11.4.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in channel 5.

(1) Example of External Pulse Width Measurement Setting Procedure

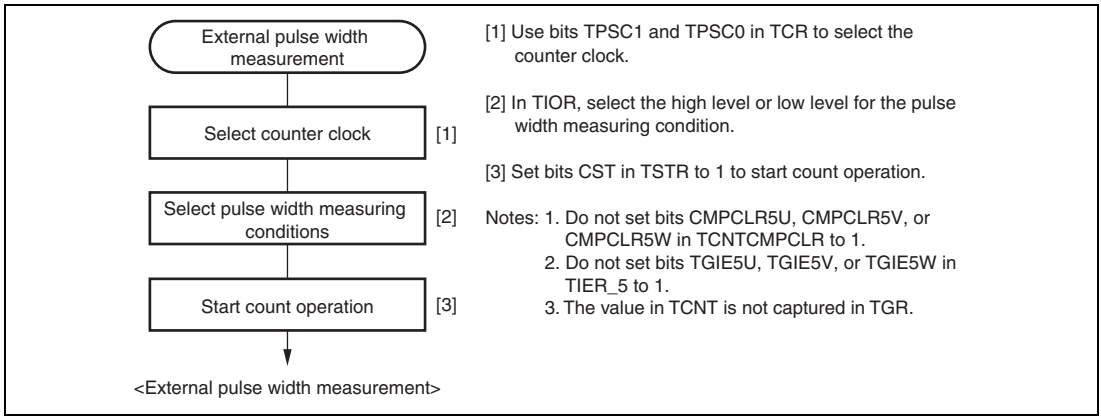


Figure 11.87 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

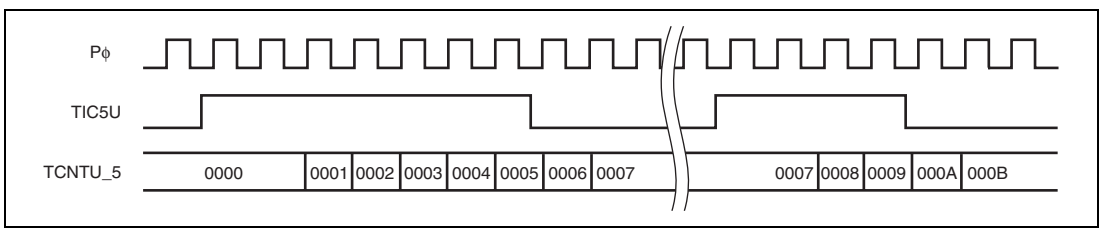


Figure 11.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)

11.4.12 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function while the complementary PWM is in operation.

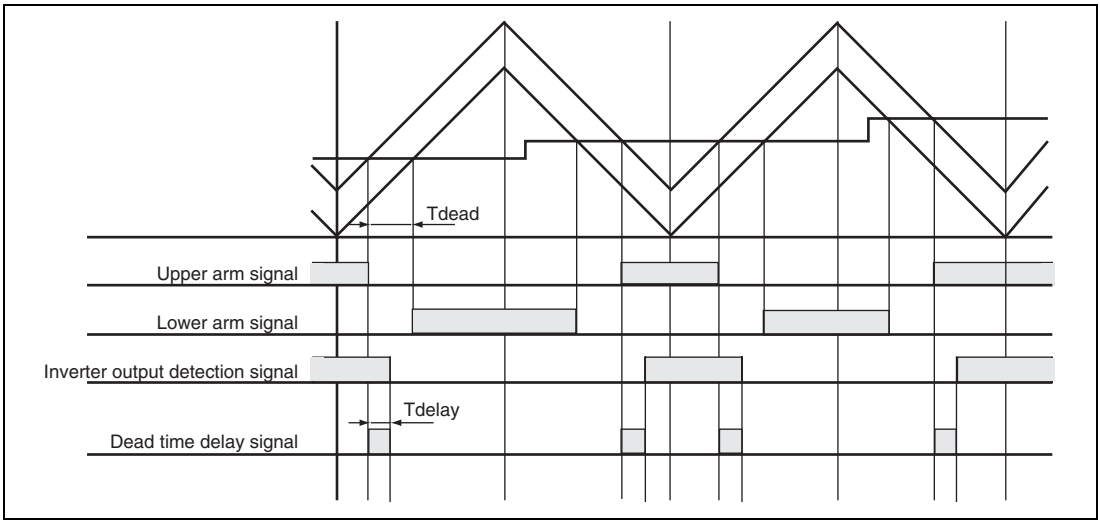
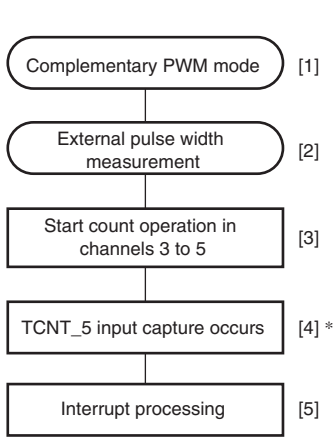


Figure 11.89 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 11.90 shows an example of dead time compensation setting procedure by using three counters in channel 5.



[1] Place channels 3 and 4 in complementary PWM mode. For details, refer to section 11.4.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in channel 5. For details, refer to section 11.4.11, External Pulse Width Measurement.

[3] Set bits CST3 and CST4 in TSTR and bits CST5U, CST5V, and CST5W in TSTR2 to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the TCNT_5 value is captured in TGR_5.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA_3) or trough (TCIV_4) in complementary PWM mode, read the TGRU_5 value, calculate the difference in time in TGRB_3, and write the corrected value to TGRD_3 in the interrupt processing.
For the V phase and W phase, read the TGRV_5 and TGRW_5 values and write the corrected values to TGRC_4 and TGRD_4, respectively, in the same way as for U-phase compensation.
The TCNT_5 value should be cleared through the TCNTCMPCLR setting or by software.

Notes: The PFC settings must be completed in advance.
* As an interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in TIER_5.

Figure 11.90 Example of Dead Time Compensation Setting Procedure

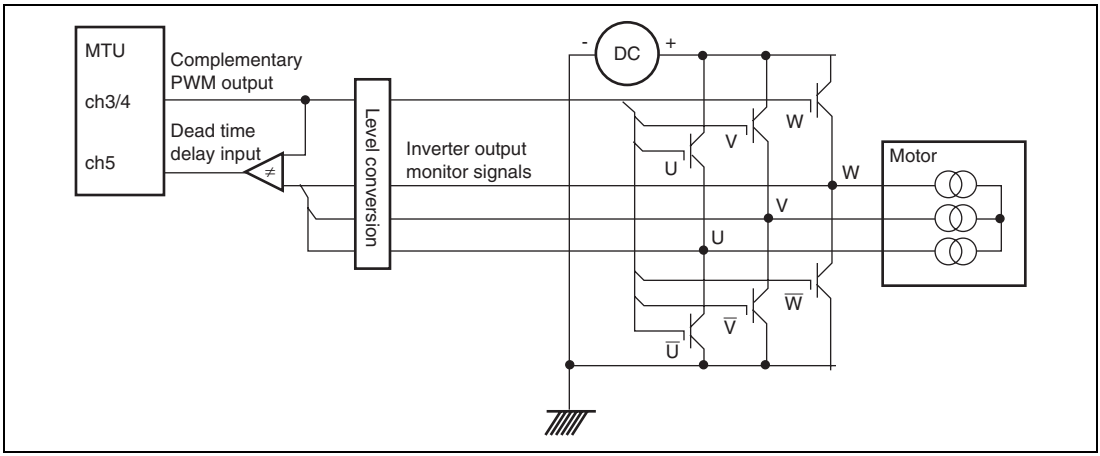


Figure 11.91 Example of Motor Control Circuit Configuration

11.4.13 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 11.92 shows an 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).

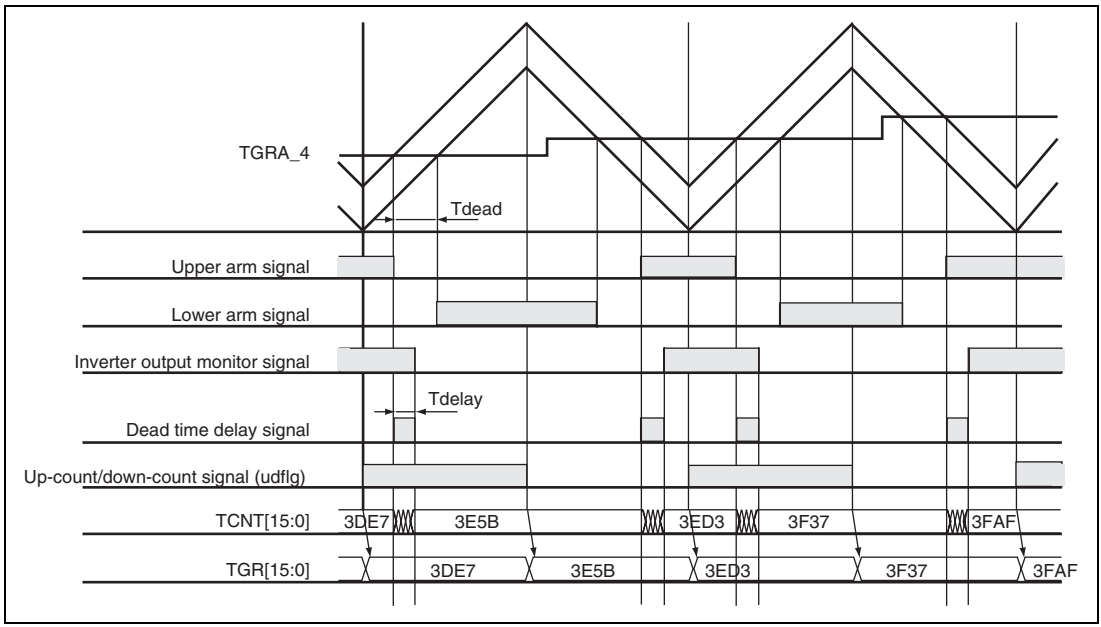


Figure 11.92 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

11.5 Interrupt Sources

11.5.1 Interrupt Sources and Priorities

There are three kinds of MTU2 interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, however the priority order within a channel is fixed. For details, see section 6, Interrupt Controller (INTC).

Table 11.57 lists the MTU2 interrupt sources.

Table 11.57 MTU2 Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DMAC Activation	Priority
0	TGIA_0	TGRA_0 input capture/compare match	TGFA_0	Possible	High
	TGIB_0	TGRB_0 input capture/compare match	TGFB_0	Not possible	
	TGIC_0	TGRC_0 input capture/compare match	TGFC_0	Not possible	
	TGID_0	TGRD_0 input capture/compare match	TGFD_0	Not possible	
	TCIV_0	TCNT_0 overflow	TCFV_0	Not possible	
	TGIE_0	TGRE_0 compare match	TGFE_0	Not possible	
	TGIF_0	TGRF_0 compare match	TGFF_0	Not possible	
1	TGIA_1	TGRA_1 input capture/compare match	TGFA_1	Possible	
	TGIB_1	TGRB_1 input capture/compare match	TGFB_1	Not possible	
	TCIV_1	TCNT_1 overflow	TCFV_1	Not possible	
	TCIU_1	TCNT_1 underflow	TCFU_1	Not possible	
2	TGIA_2	TGRA_2 input capture/compare match	TGFA_2	Possible	
	TGIB_2	TGRB_2 input capture/compare match	TGFB_2	Not possible	
	TCIV_2	TCNT_2 overflow	TCFV_2	Not possible	
	TCIU_2	TCNT_2 underflow	TCFU_2	Not possible	
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible	
	TGIB_3	TGRB_3 input capture/compare match	TGFB_3	Not possible	
	TGIC_3	TGRC_3 input capture/compare match	TGFC_3	Not possible	
	TGID_3	TGRD_3 input capture/compare match	TGFD_3	Not possible	
	TCIV_3	TCNT_3 overflow	TCFV_3	Not possible	
4	TGIA_4	TGRA_4 input capture/compare match	TGFA_4	Possible	
	TGIB_4	TGRB_4 input capture/compare match	TGFB_4	Not possible	
	TGIC_4	TGRC_4 input capture/compare match	TGFC_4	Not possible	
	TGID_4	TGRD_4 input capture/compare match	TGFD_4	Not possible	
	TCIV_4	TCNT_4 overflow/underflow	TCFV_4	Not possible	
5	TGIU_5	TGRU_5 input capture/compare match	TGFU_5	Not possible	
	TGIV_5	TGRV_5 input capture/compare match	TGFV_5	Not possible	
	TGIW_5	TGRW_5 input capture/compare match	TGFW_5	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The MTU2 has 21 input capture/compare match interrupts, six for channel 0, four each for channels 3 and 4, two each for channels 1 and 2, and three for channel 5. The TGFE_0 and TGFF_0 flags in channel 0 are not set by the occurrence of an input capture.

(2) Overflow Interrupt

An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The MTU2 has five overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The MTU2 has two underflow interrupts, one each for channels 1 and 2.

11.5.2 DMAC and DTC Activation

(1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel and the overflow interrupt of channel 4. For details, see section 8, Data Transfer Controller (DTC).

In the MTU2, a total of twenty input capture/compare match interrupts and overflow interrupts 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.

(2) DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 10, Direct Memory Access Controller (DMAC).

In the MTU2, a total of five TGRA input capture/compare match interrupts can be used as DMAC activation sources, one each for channels 0 to 4.

When the DMAC is activation by MTU2, the activation sources are cleared when the DMAC requests the internal bus mastership. Accordingly, depending on the internal bus state, a wait state

of the DMAC transfer may be generated even if the activation sources are cleared. Also, when transferring DMAC burst by MTU2, the setting of bus function extension register (BSCEHR) is required. See section 9.4.8, Bus Function Extending Register (BSCEHR), for details.

11.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 11.58 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 TCNT_4 Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in TIER_4 is set to 1, the A/D converter can be activated at the trough of TCNT_4 count (TCNT_4 = H'0000).

A/D converter start request signal TRGAN is issued to the A/D converter under either one of the following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel while the TTGE bit in TIER is set to 1
- When the TCNT_4 count reaches the trough (TCNT_4 = H'0000) during complementary PWM operation while the TTGE2 bit in TIER_4 is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between TCNT_0 and TGRE_0

The A/D converter can be activated by generating A/D converter start request signal TRG0N when a compare match occurs between TCNT_0 and TGRE_0 in channel 0.

When the TGFE flag in TSR2_0 is set to 1 by the occurrence of a compare match between TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0N from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 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 11.4.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4AN from the MTU2 is selected as the trigger in the A/D converter when TRG4AN is generated or if TRG4BN from the MTU2 is selected as the trigger in the A/D converter when TRG4BN is generated.

Table 11.58 Interrupt Sources and A/D Converter Start Request Signals

Target Registers	Interrupt Source	A/D Converter Start Request Signal
TGRA_0 and TCNT_0	Input capture/compare match	TRGAN
TGRA_1 and TCNT_1		
TGRA_2 and TCNT_2		
TGRA_3 and TCNT_3		
TGRA_4 and TCNT_4		
TCNT_4	TCNT_4 Trough in complementary PWM mode	
TGRE_0 and TCNT_0	Compare match	TRG0N
TADCORA and TCNT_4		TRG4AN
TADCORB and TCNT_4		TRG4BN

11.6 Operation Timing

11.6.1 Input/Output Timing

(1) TCNT Count Timing

Figures 11.93 and 94 show TCNT count timing in internal clock operation, and figure 11.95 shows TCNT count timing in external clock operation (normal mode), and figure 11.96 shows TCNT count timing in external clock operation (phase counting mode).

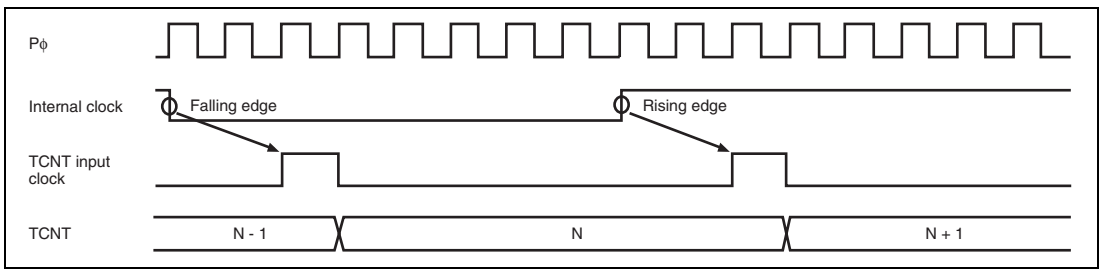


Figure 11.93 Count Timing in Internal Clock Operation (Channels 0 to 4)

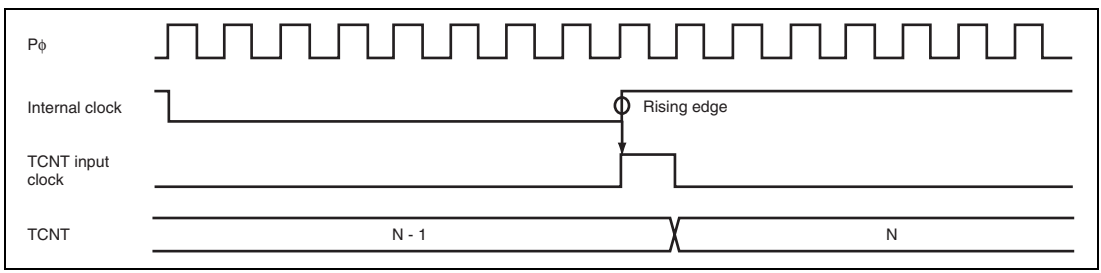


Figure 11.94 Count Timing in Internal Clock Operation (Channel 5)

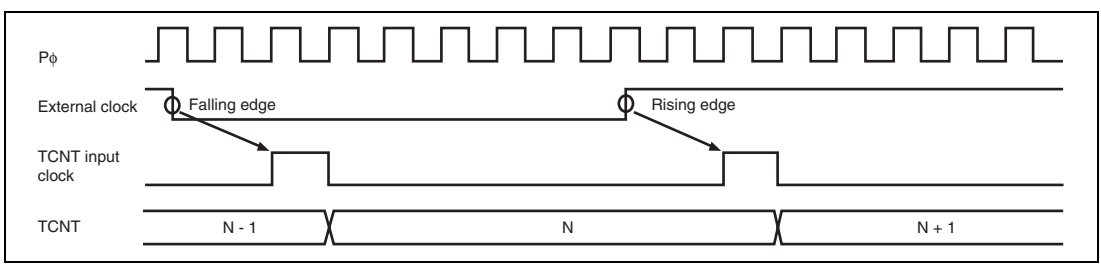


Figure 11.95 Count Timing in External Clock Operation (Channels 0 to 4)

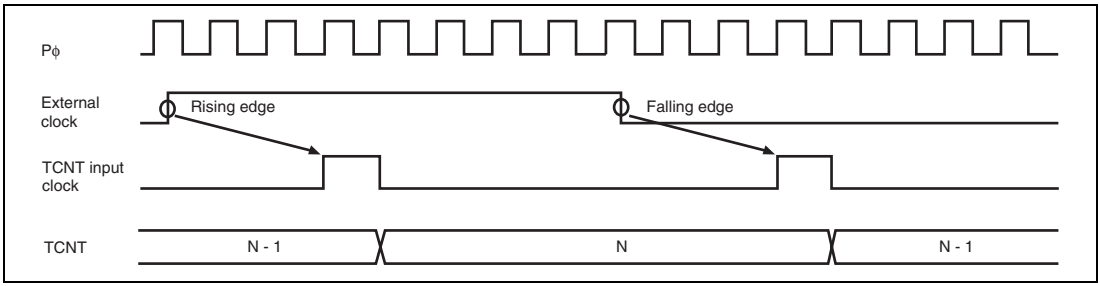


Figure 11.96 Count Timing in External Clock Operation (Phase Counting Mode)

(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 output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 11.97 shows output compare output timing (normal mode and PWM mode) and figure 11.98 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

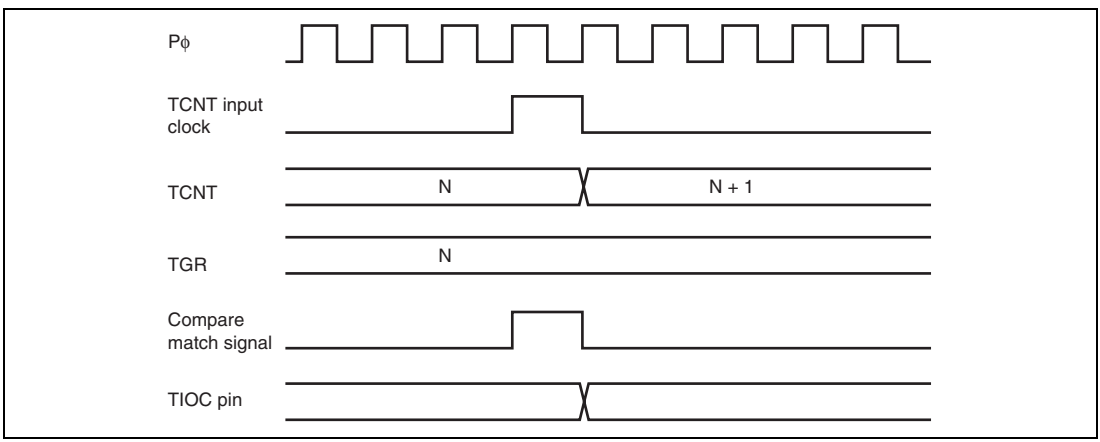


Figure 11.97 Output Compare Output Timing (Normal Mode/PWM Mode)

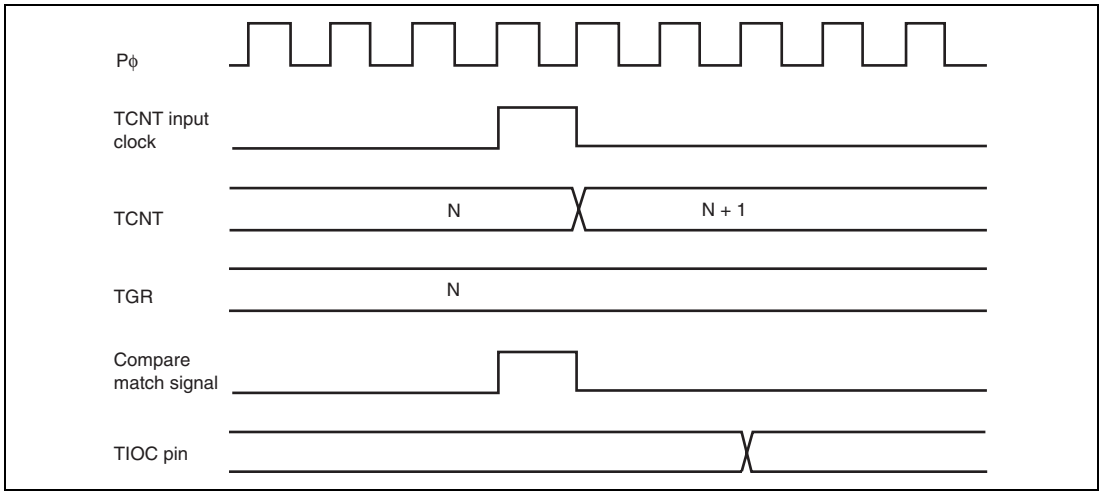


Figure 11.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 11.99 shows input capture signal timing.

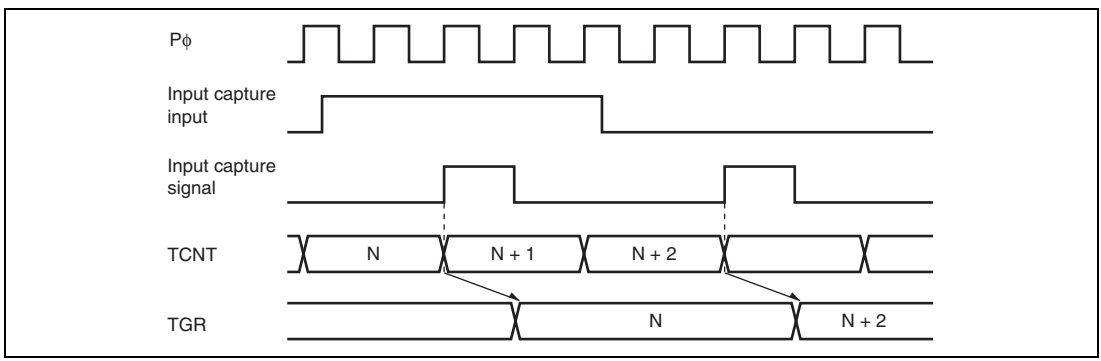


Figure 11.99 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figures 11.100 and 101 show the timing when counter clearing on compare match is specified, and figure 11.102 shows the timing when counter clearing on input capture is specified.

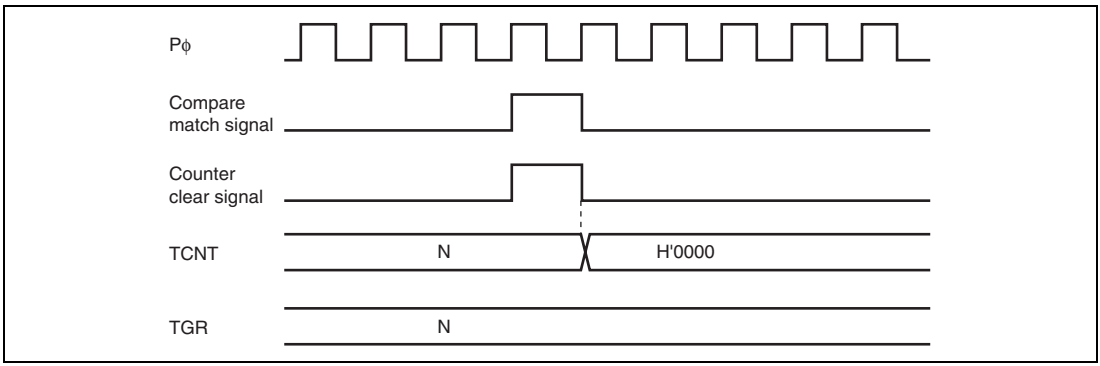


Figure 11.100 Counter Clear Timing (Compare Match) (Channels 0 to 4)

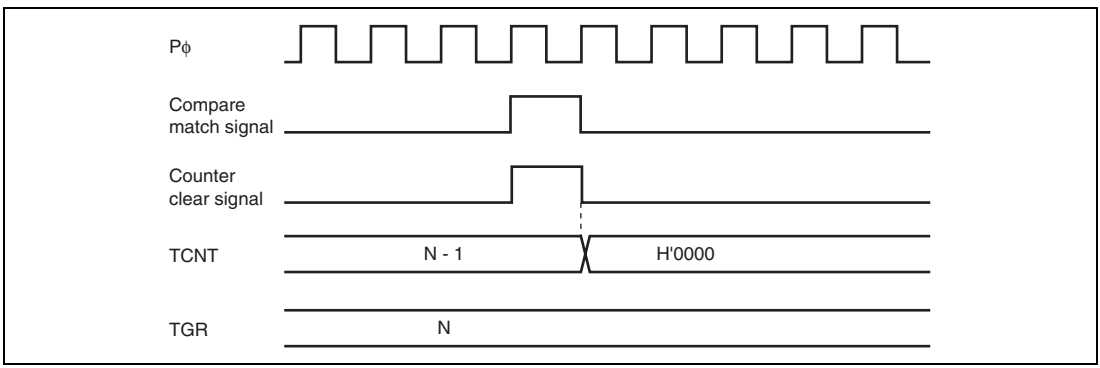


Figure 11.101 Counter Clear Timing (Compare Match) (Channel 5)

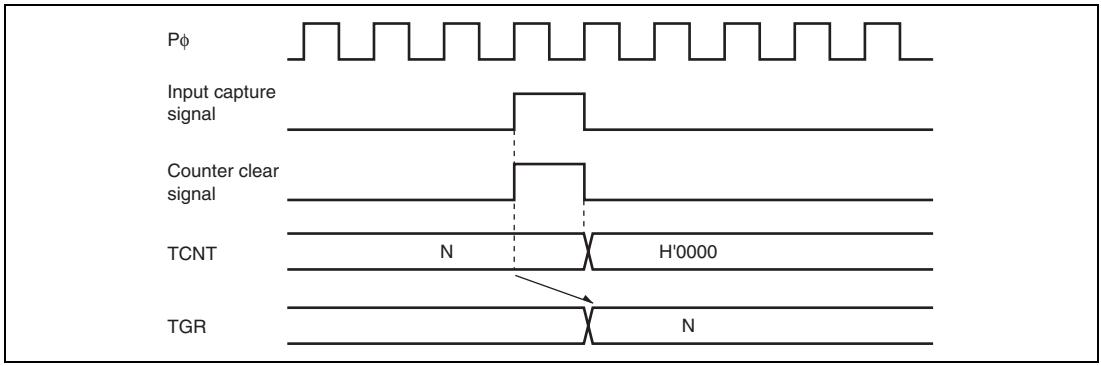


Figure 11.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)

(5) Buffer Operation Timing

Figures 11.103 to 11.105 show the timing in buffer operation.

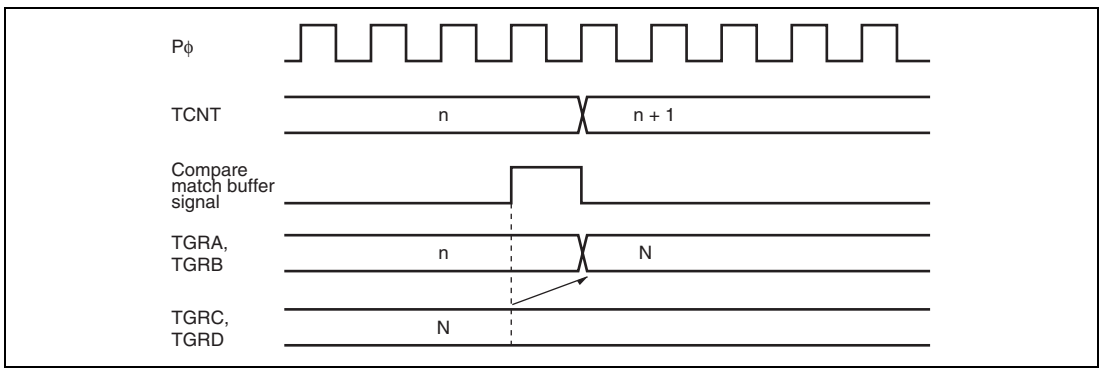


Figure 11.103 Buffer Operation Timing (Compare Match)

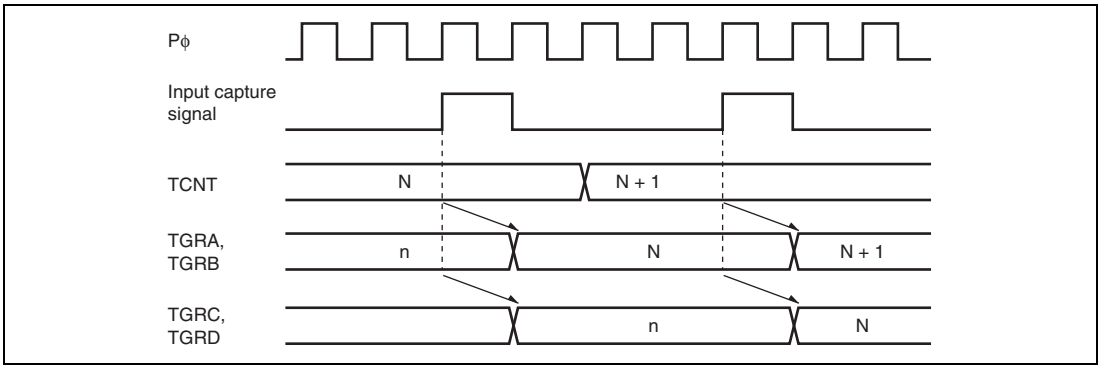


Figure 11.104 Buffer Operation Timing (Input Capture)

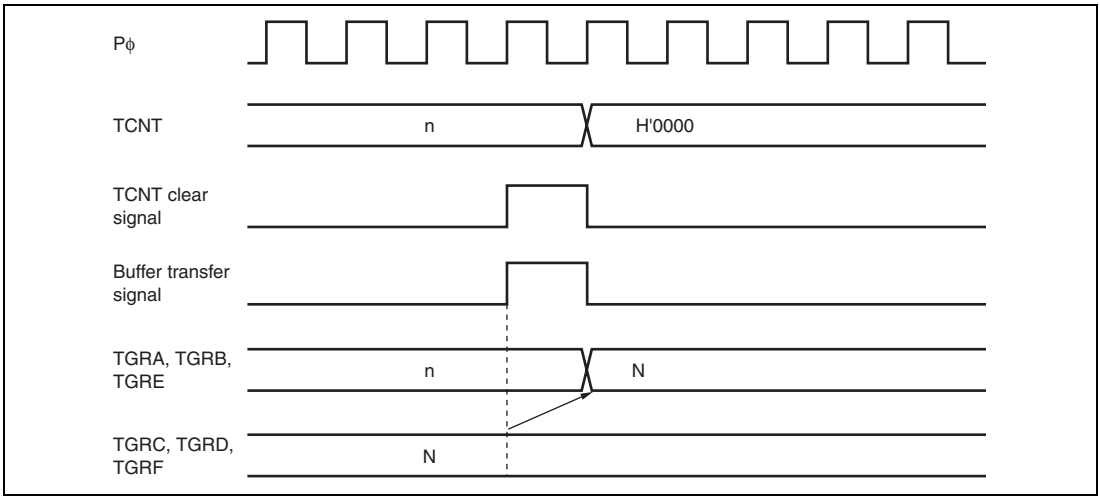


Figure 11.105 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 11.106 to 11.108 show the buffer transfer timing in complementary PWM mode.

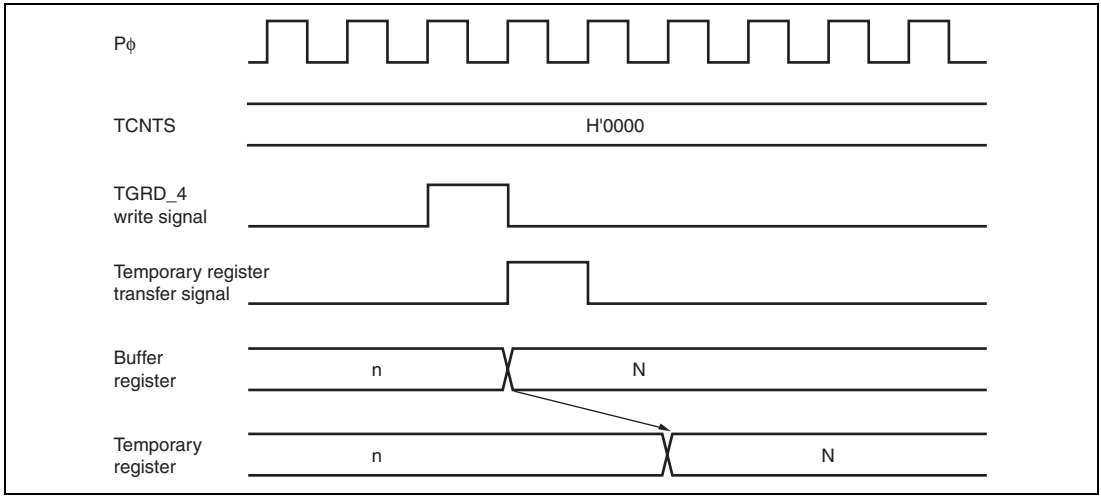


Figure 11.106 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

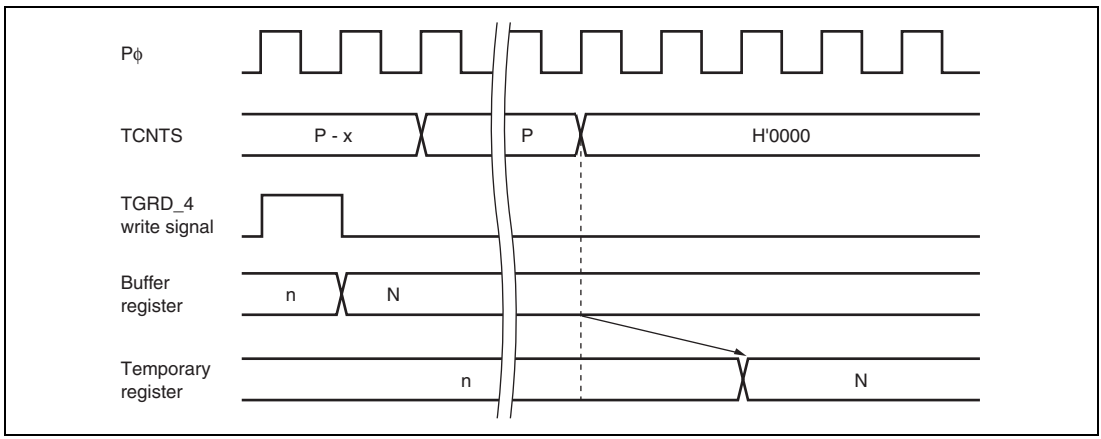


Figure 11.107 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

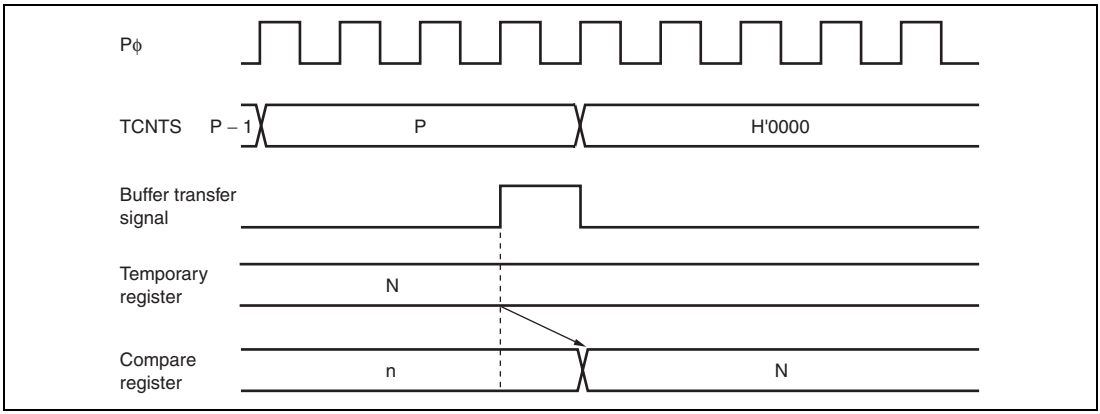


Figure 11.108 Transfer Timing from Temporary Register to Compare Register

11.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figures 11.109 and 110 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

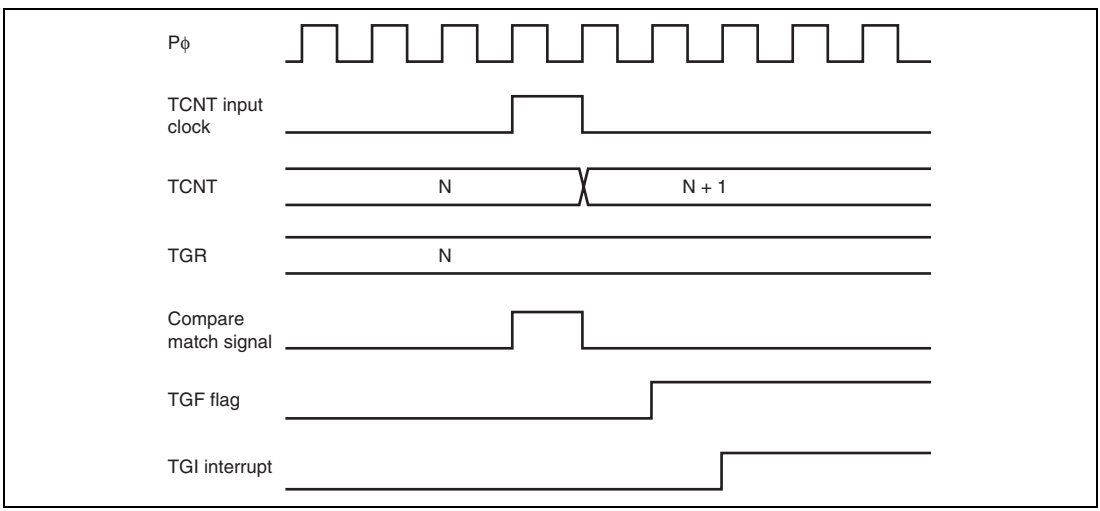


Figure 11.109 TGI Interrupt Timing (Compare Match)

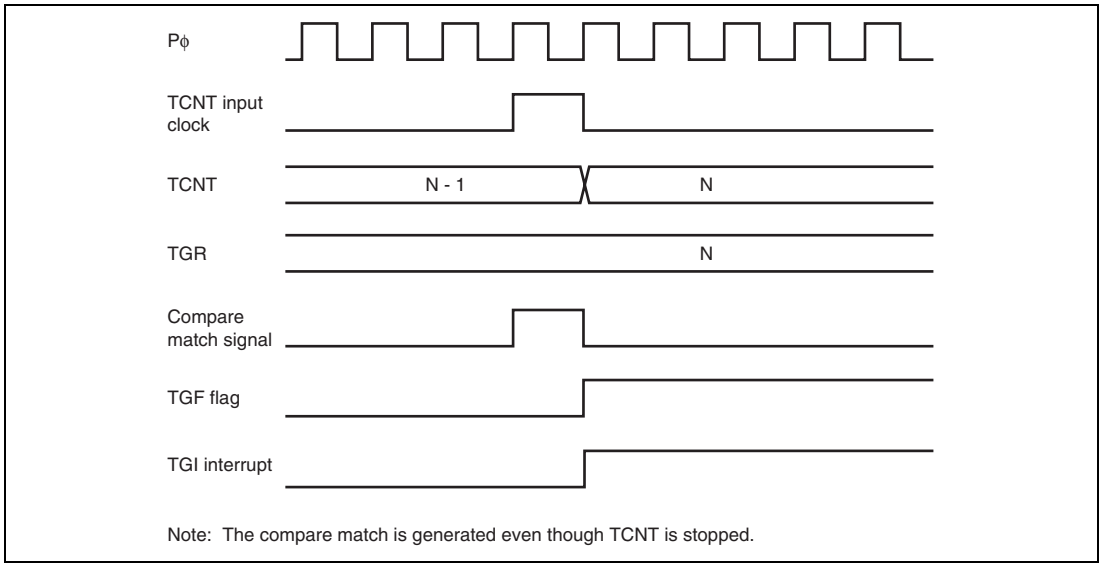


Figure 11.110 TGI Interrupt Timing (Compare Match) (Channel 5)

(2) TGF Flag Setting Timing in Case of Input Capture

Figures 11.111 and 112 show the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

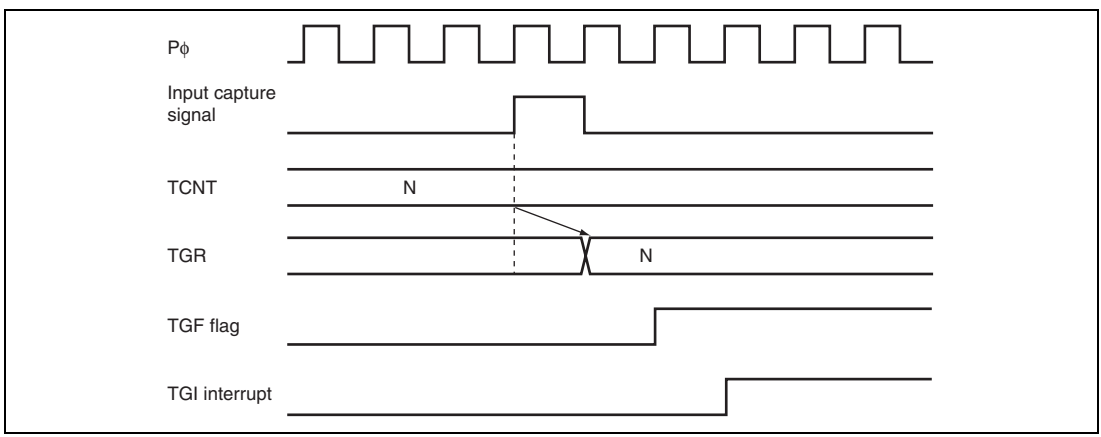


Figure 11.111 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)

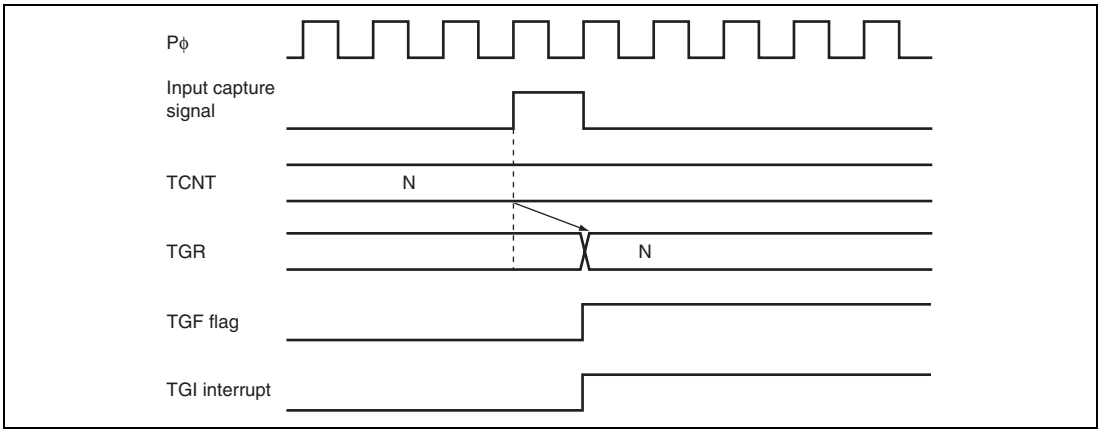


Figure 11.112 TGI Interrupt Timing (Input Capture) (Channel 5)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 11.113 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 11.114 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

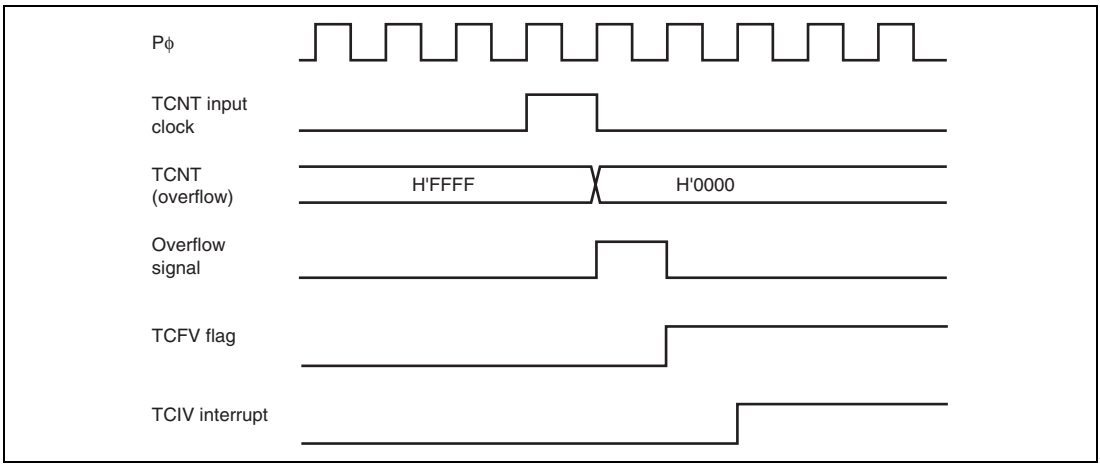


Figure 11.113 TCIV Interrupt Setting Timing

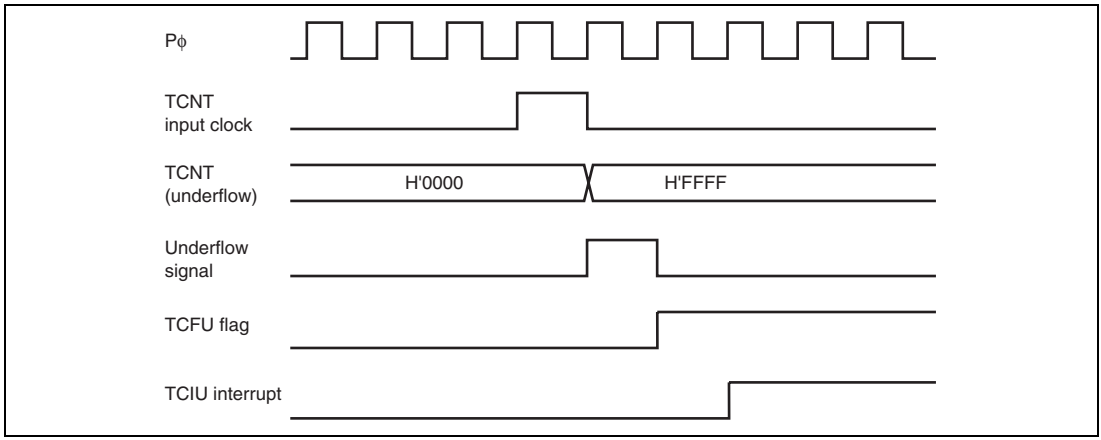


Figure 11.114 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DMAC is activated, the flag is cleared automatically. Figures 11.115 and 116 show the timing for status flag clearing by the CPU, and figure 11.117 shows the timing for status flag clearing by the DMAC.

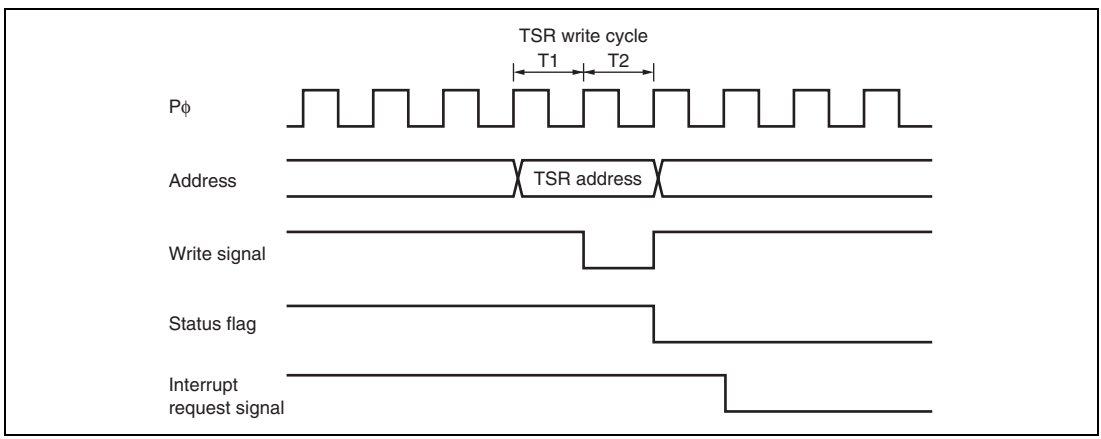


Figure 11.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)

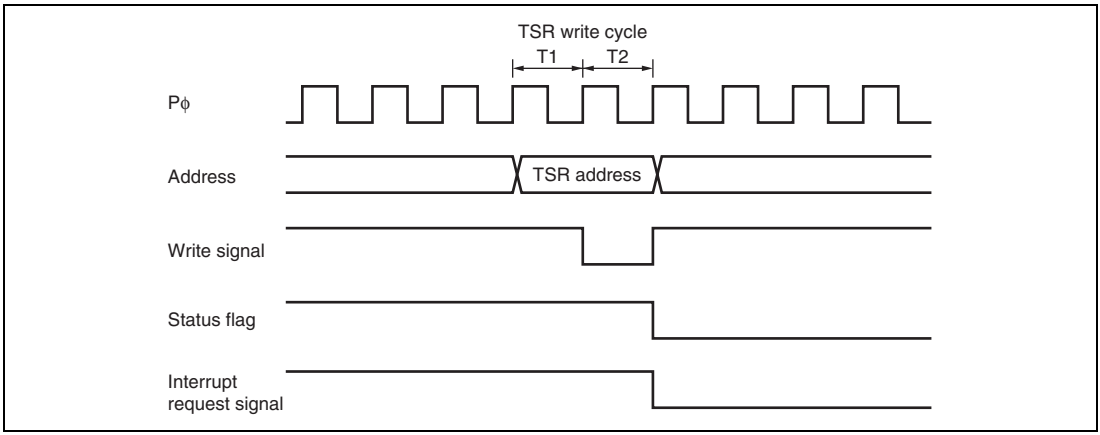


Figure 11.116 Timing for Status Flag Clearing by CPU (Channel 5)

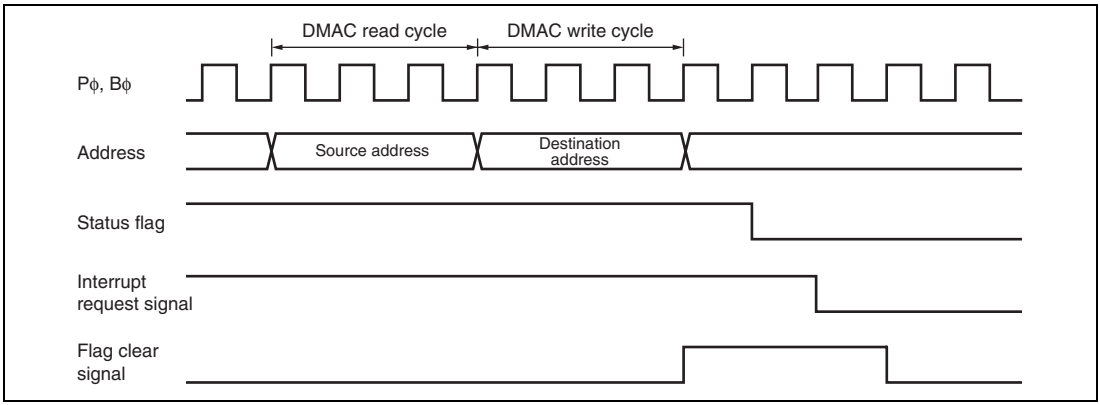


Figure 11.117 Timing for Status Flag Clearing by DTC Activation (Channels 0 to 4)

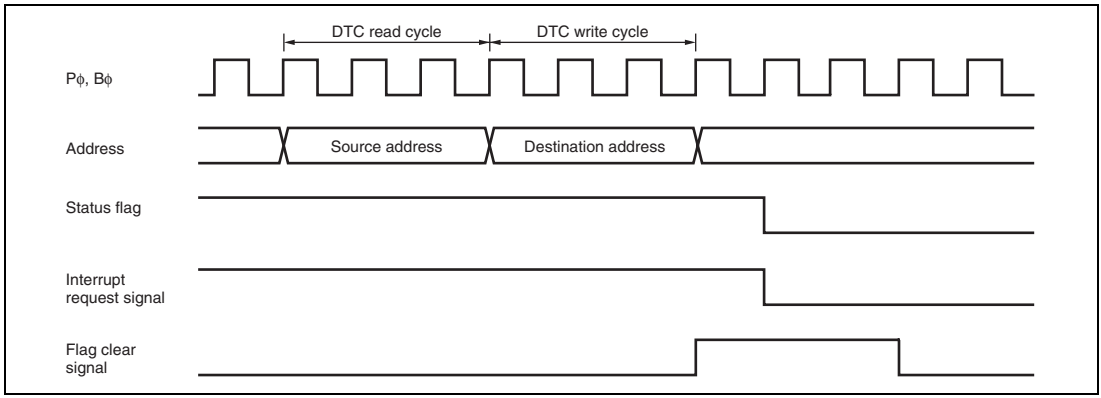


Figure 11.118 Timing for Status Flag Clearing by DTC Activation (Channel 5)

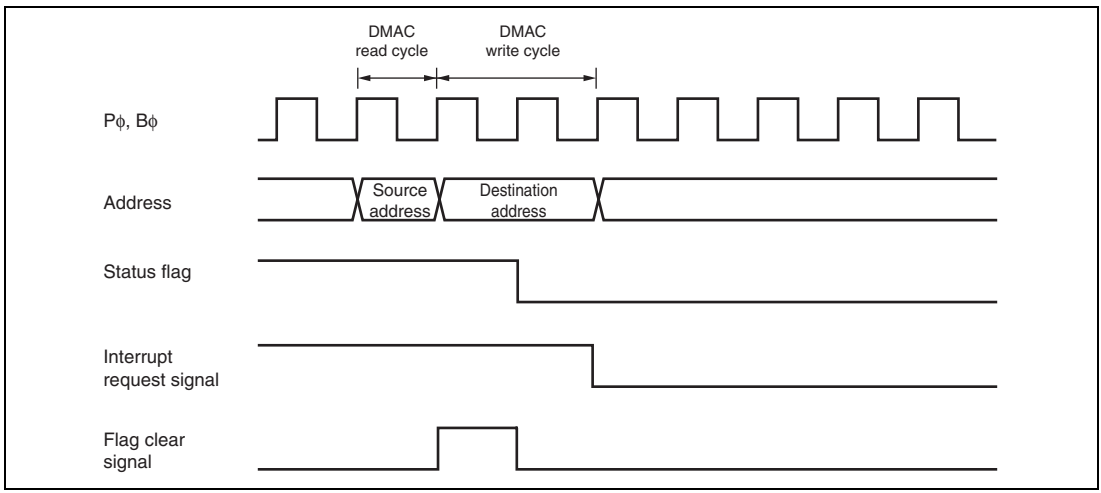


Figure 11.119 Timing for Status Flag Clearing by DMAC Activation

11.7 Usage Notes

11.7.1 Module Standby Mode Setting

MTU2 operation can be disabled or enabled using the standby control register. The initial setting is for MTU2 operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 30, Power-Down Modes.

11.7.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The MTU2 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 11.120 shows the input clock conditions in phase counting mode.

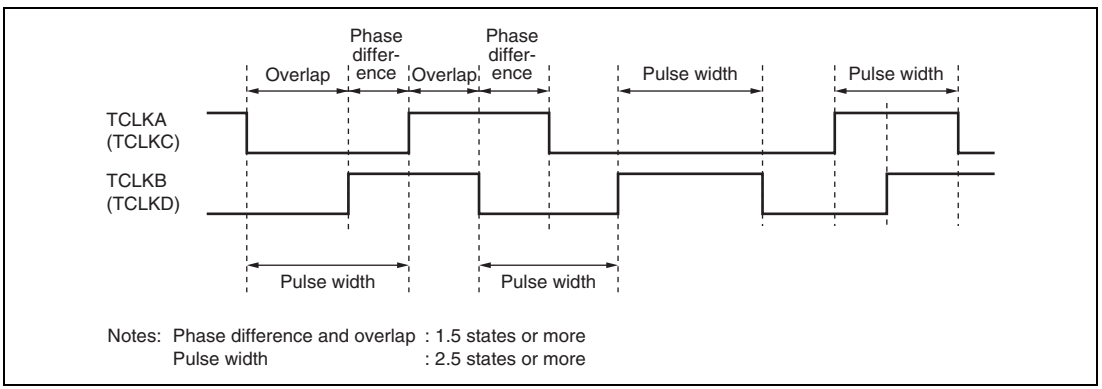


Figure 11.120 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

11.7.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

- Channel 0 to 4

$$f = \frac{P\phi}{(N + 1)}$$

- Channel 5

$$f = \frac{P\phi}{N}$$

Where f: Counter frequency
Pϕ: Peripheral clock operating frequency
N: TGR set value

11.7.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 11.121 shows the timing in this case.

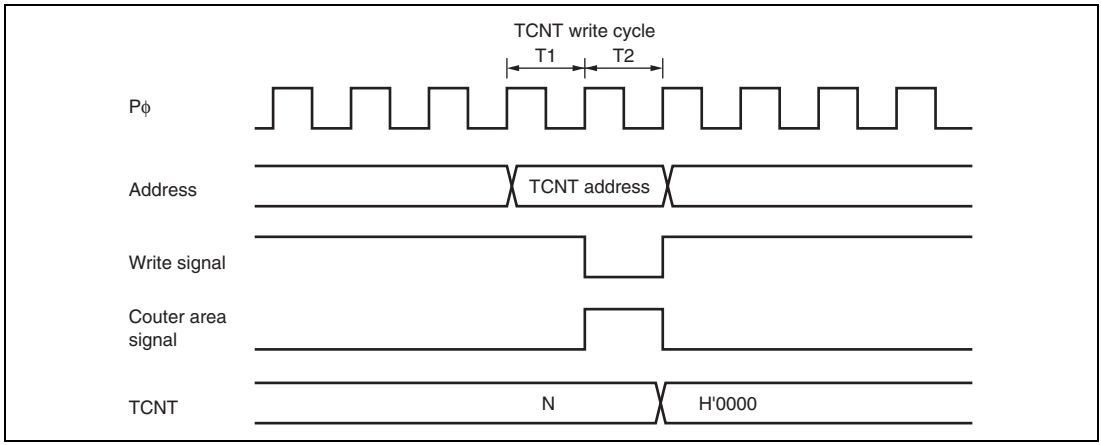


Figure 11.121 Contention between TCNT Write and Clear Operations

11.7.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 11.122 shows the timing in this case.

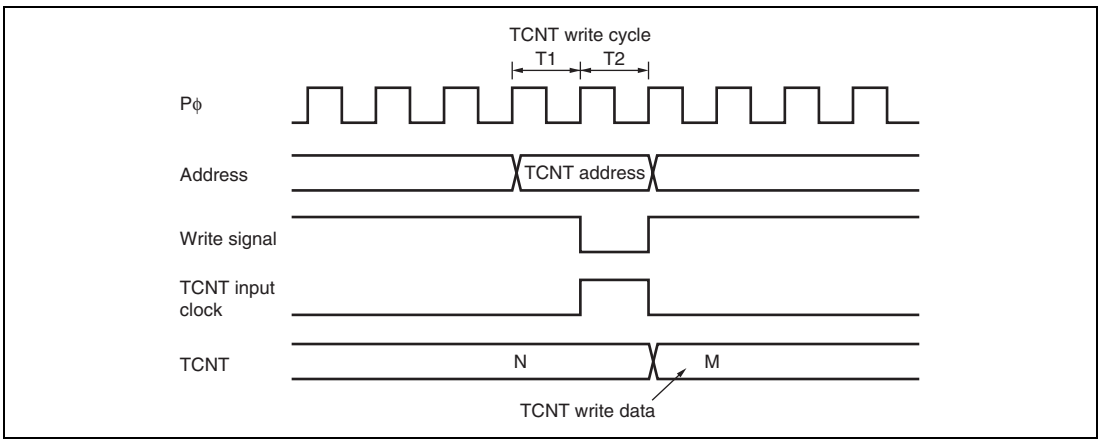


Figure 11.122 Contention between TCNT Write and Increment Operations

11.7.6 Contention between TGR Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the TGR write is executed and the compare match signal is also generated.

Figure 11.123 shows the timing in this case.

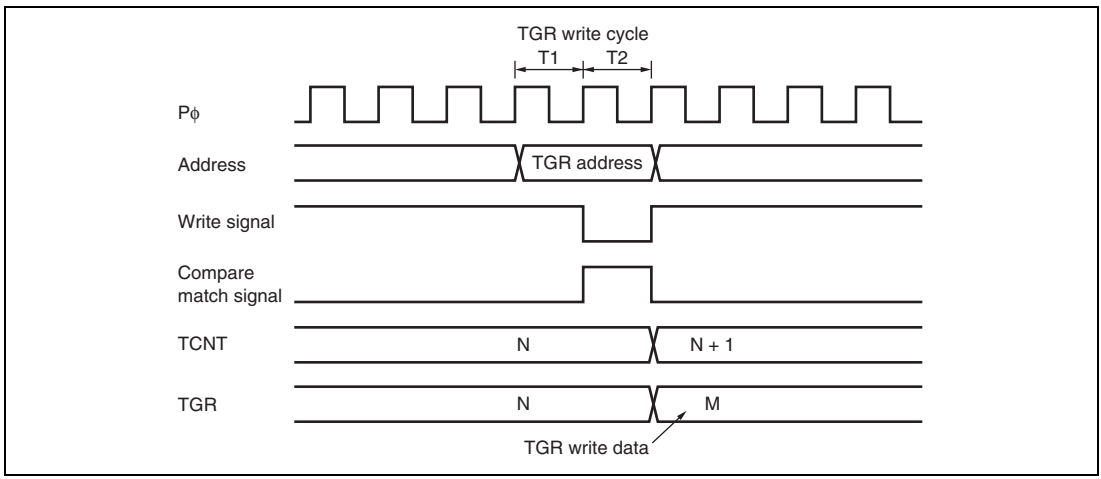


Figure 11.123 Contention between TGR Write and Compare Match

11.7.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data after write.

Figure 11.124 shows the timing in this case.

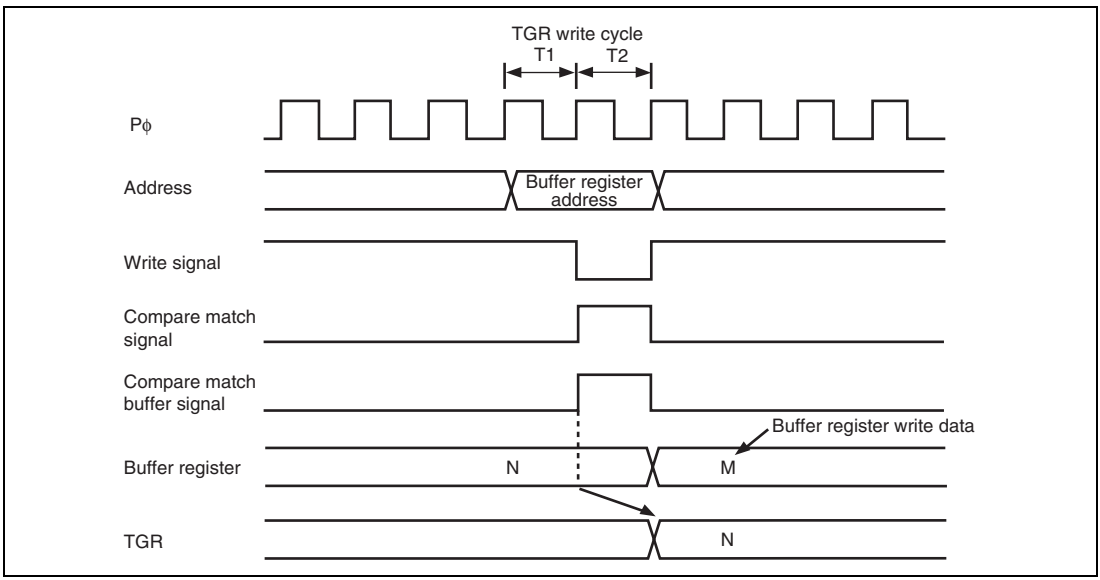


Figure 11.124 Contention between Buffer Register Write and Compare Match

11.7.8 Contention between Buffer Register Write and TCNT Clear

When the buffer transfer timing is set at the TCNT clear by the buffer transfer mode register (TBTM), if TCNT clear occurs in the T2 state of a TGR write cycle, the data that is transferred to TGR by the buffer operation is the data before write.

Figure 11.125 shows the timing in this case.

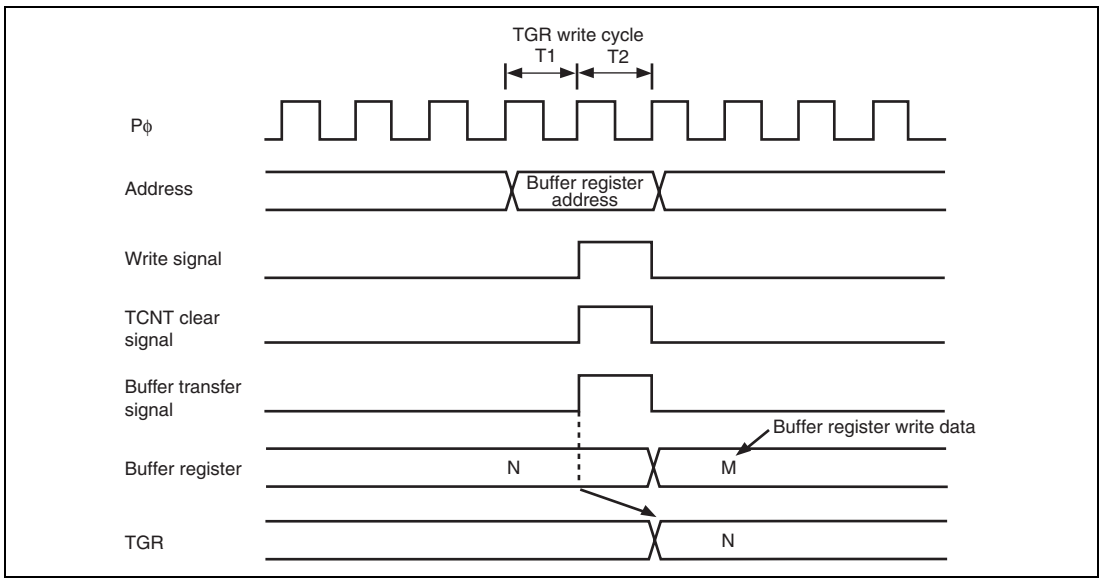


Figure 11.125 Contention between Buffer Register Write and TCNT Clear

11.7.9 Contention between TGR Read and Input Capture

If an input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data in the buffer before input capture transfer for channels 0 to 4, and the data after input capture transfer for channel 5.

Figures 11.126 and 127 show the timing in this case.

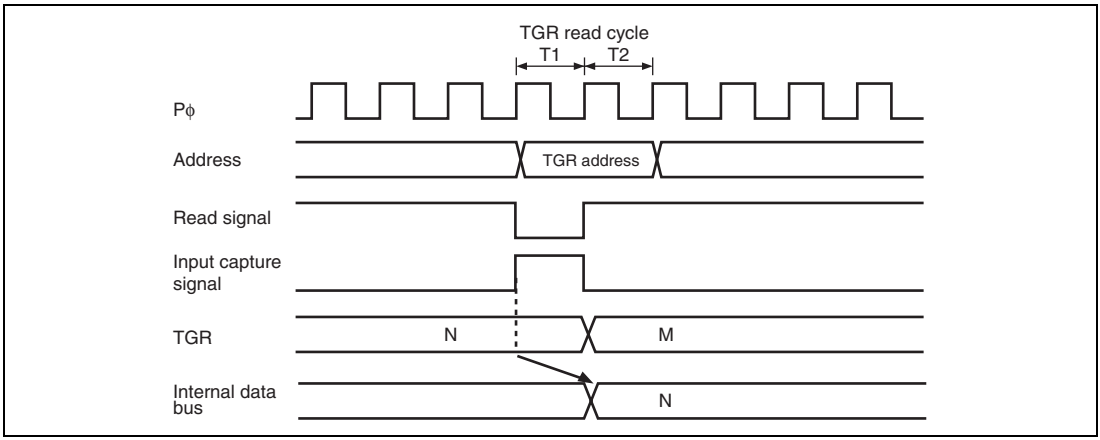


Figure 11.126 Contention between TGR Read and Input Capture (Channels 0 to 4)

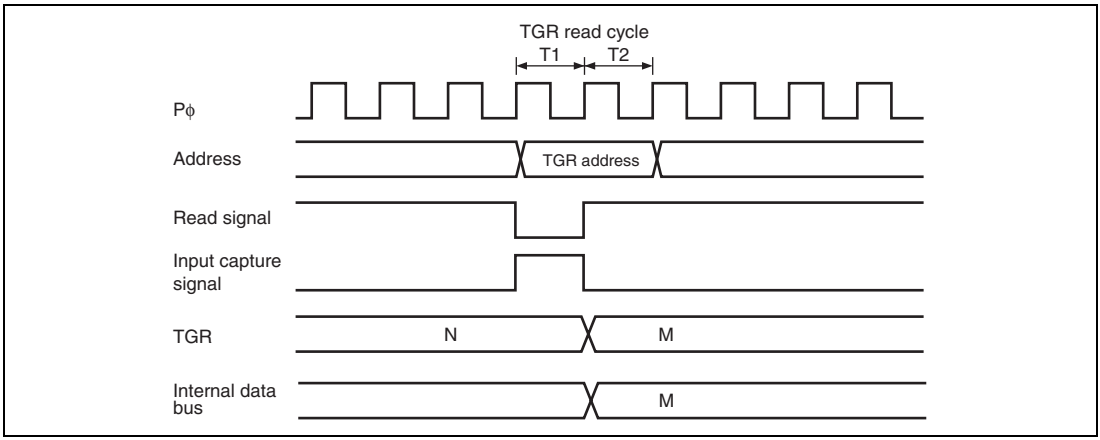


Figure 11.127 Contention between TGR Read and Input Capture (Channel 5)

11.7.10 Contention between TGR Write and Input Capture

If an input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed for channels 0 to 4. For channel 5, write to TGR is performed and the input capture signal is generated.

Figures 11.128 and 129 show the timing in this case.

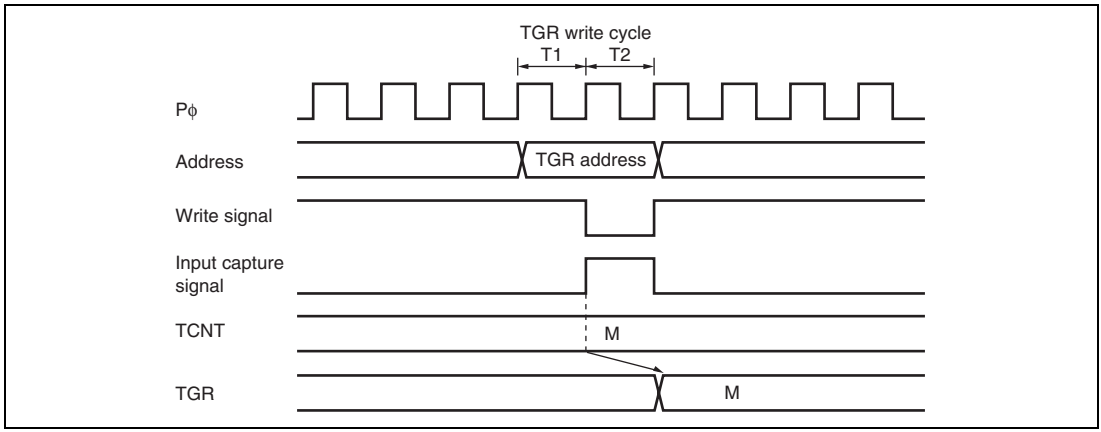


Figure 11.128 Contention between TGR Write and Input Capture (Channels 0 to 4)

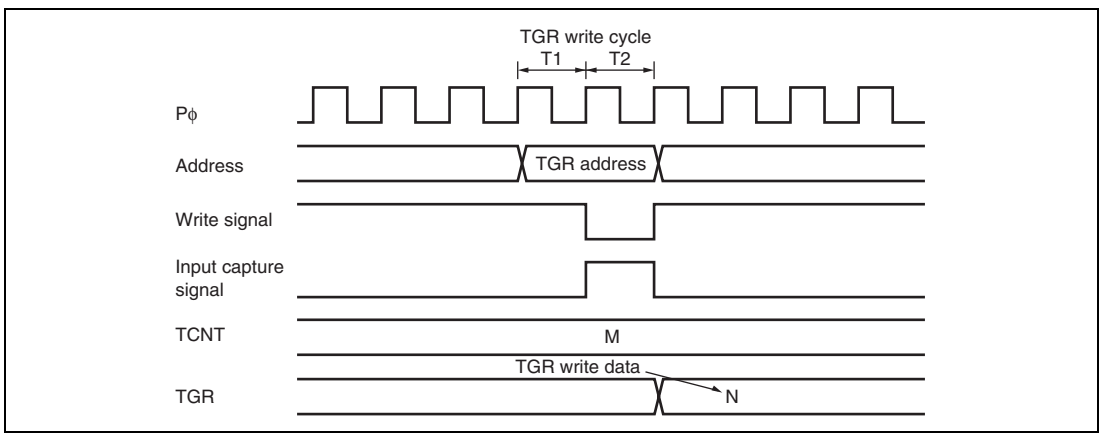


Figure 11.129 Contention between TGR Write and Input Capture (Channel 5)

11.7.11 Contention between Buffer Register Write and Input Capture

If an input capture signal is generated in the T2 state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 11.130 shows the timing in this case.

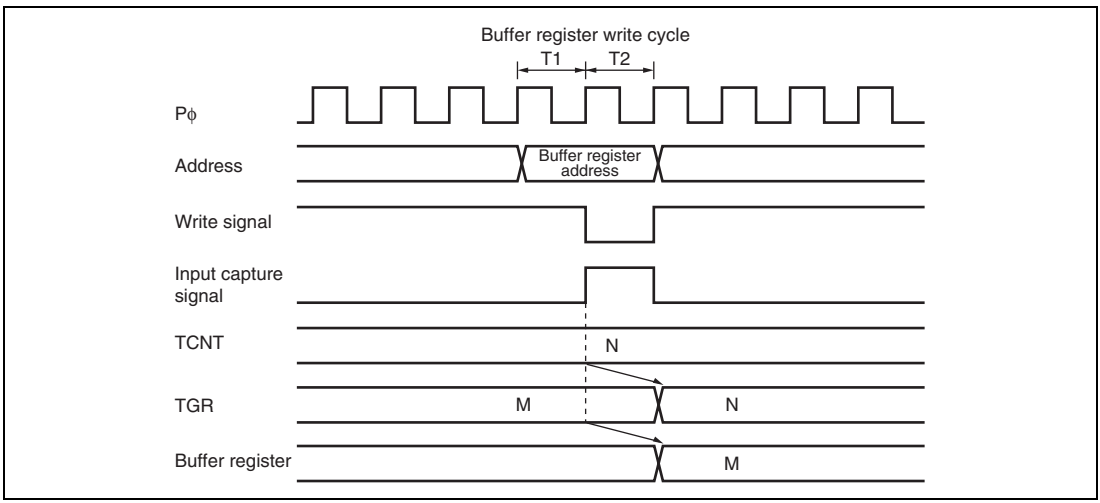


Figure 11.130 Contention between Buffer Register Write and Input Capture

11.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT1 and TCNT2 in a cascade connection, when a contention occurs during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T_2 state of the TCNT_2 write cycle, the write to TCNT_2 is conducted, and the TCNT_1 count signal is disabled. At this point, if there is match with TGRA_1 and the TCNT_1 value, a compare signal is issued. Furthermore, when the TCNT_1 count clock is selected as the input capture source of channel 0, TGRA_0 to D_0 carry out the input capture operation. In addition, when the compare match/input capture is selected as the input capture source of TGRB_1, TGRB_1 carries out input capture operation. The timing is shown in figure 11.131.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

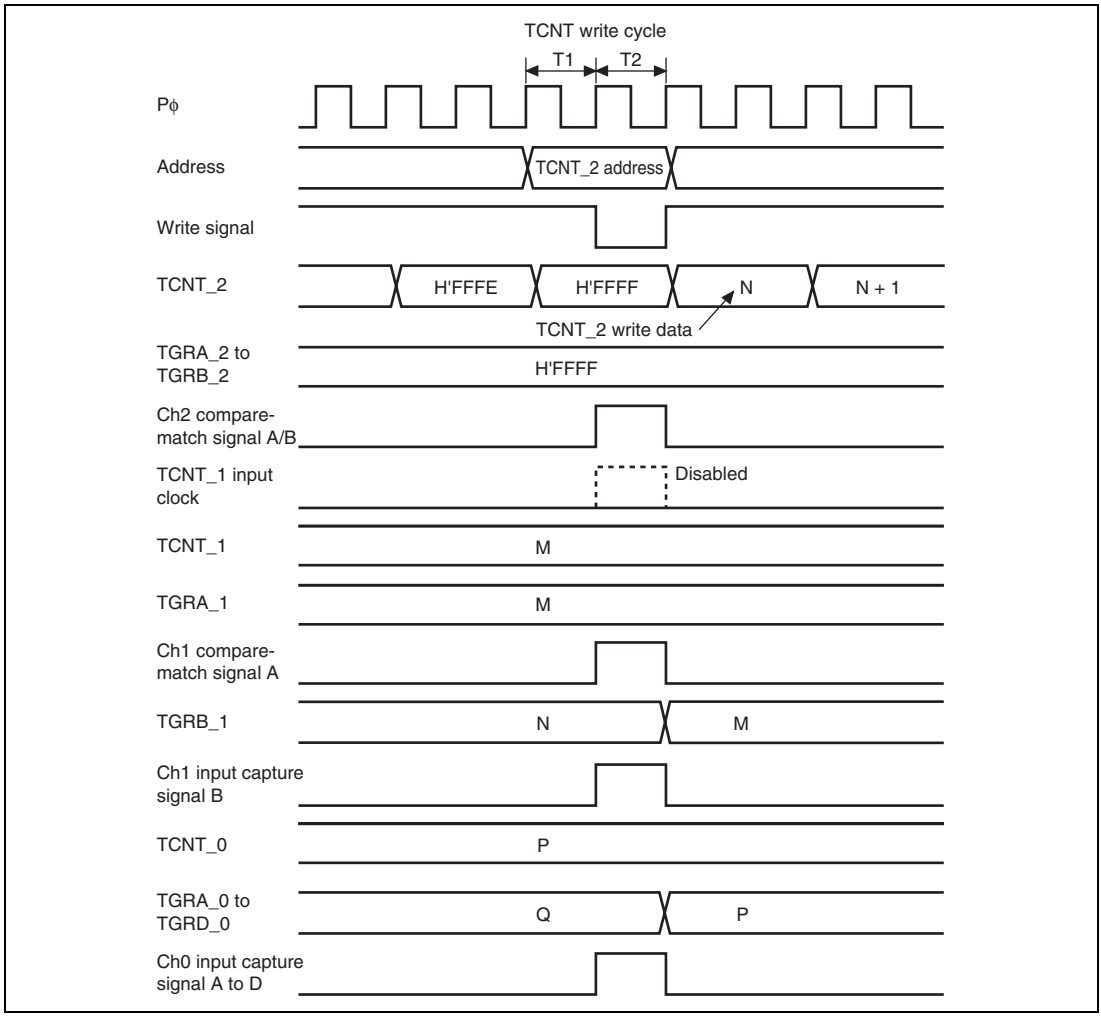


Figure 11.131 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

11.7.13 Counter Value during Complementary PWM Mode Stop

When counting operation is suspended with TCNT_3 and TCNT_4 in complementary PWM mode, TCNT_3 has the timer dead time register (TDDR) value, and TCNT_4 is held at H'0000.

When restarting complementary PWM mode, counting begins automatically from the initialized state. This explanatory diagram is shown in figure 11.132.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

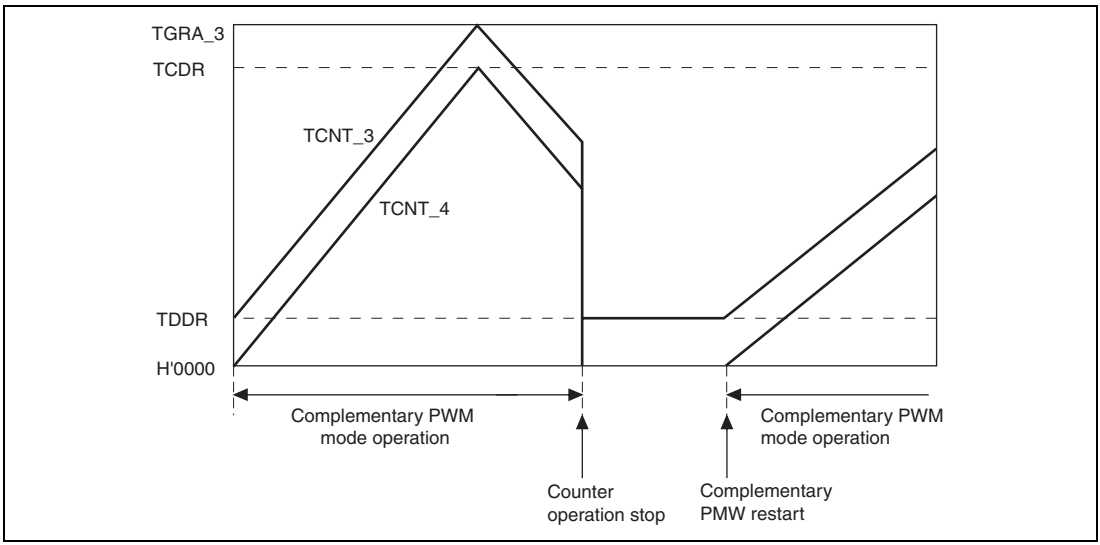


Figure 11.132 Counter Value during Complementary PWM Mode Stop

11.7.14 Buffer Operation Setting in Complementary PWM Mode

In complementary PWM mode, conduct rewrites by buffer operation for the PWM cycle setting register (TGRA_3), timer cycle data register (TCDR), and duty setting registers (TGRB_3, TGRA_4, and TGRB_4).

In complementary PWM mode, channel 3 and channel 4 buffers operate in accordance with bit settings BFA and BFB of TMDR_3. When TMDR_3's BFA bit is set to 1, TGRC_3 functions as a buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4, and TCBR functions as the TCDR's buffer register.

11.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag

When setting buffer operation for reset sync PWM mode, set the BFA and BFB bits of TMDR_4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit of TMDR_4 is set to 1.

In reset sync PWM mode, the channel 3 and channel 4 buffers operate in accordance with the BFA and BFB bit settings of TMDR_3. For example, if the BFA bit of TMDR_3 is set to 1, TGRC_3 functions as the buffer register for TGRA_3. At the same time, TGRC_4 functions as the buffer register for TGRA_4.

The TGFC bit and TGFD bit of TSR_3 and TSR_4 are not set when TGRC_3 and TGRD_3 are operating as buffer registers.

Figure 11.133 shows an example of operations for TGR_3, TGR_4, TIOC3, and TIOC4, with TMDR_3's BFA and BFB bits set to 1, and TMDR_4's BFA and BFB bits set to 0.

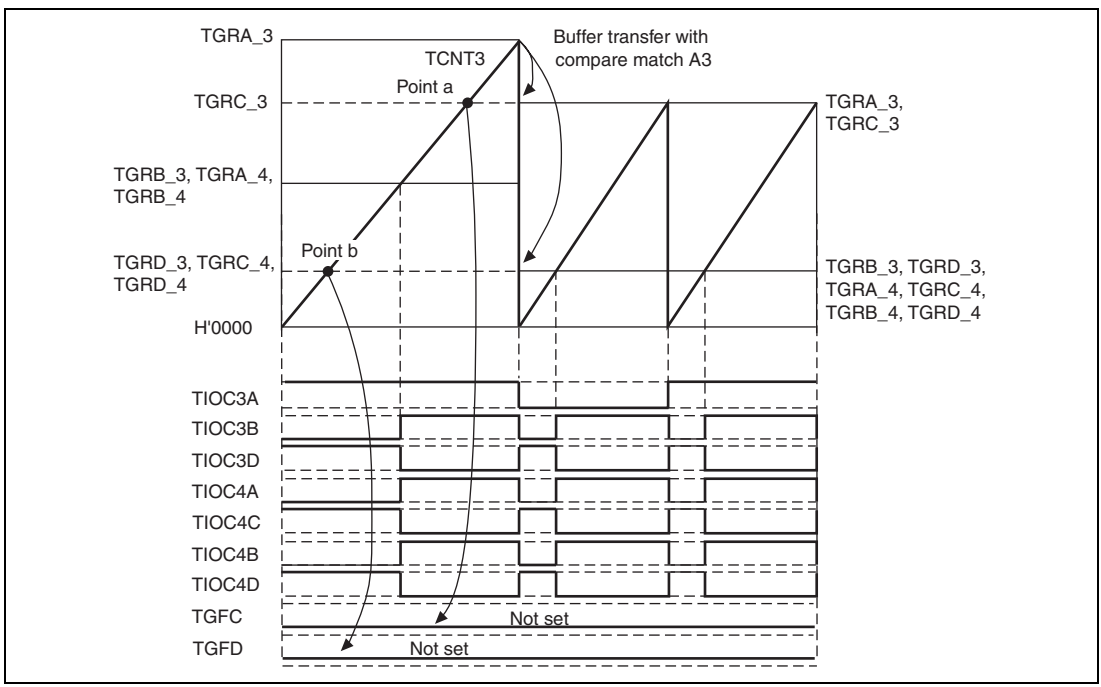


Figure 11.133 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

11.7.16 Overflow Flags in Reset Synchronous PWM Mode

When set to reset synchronous PWM mode, TCNT_3 and TCNT_4 start counting when the CST3 bit of TSTR is set to 1. At this point, TCNT_4's count clock source and count edge obey the TCR_3 setting.

In reset synchronous PWM mode, with cycle register TGRA_3's set value at H'FFFF, when specifying TGR3A compare-match for the counter clear source, TCNT_3 and TCNT_4 count up to H'FFFF, then a compare-match occurs with TGRA_3, and TCNT_3 and TCNT_4 are both cleared. At this point, TSR's overflow flag TCFV bit is not set.

Figure 11.134 shows a TCFV bit operation example in reset synchronous PWM mode with a set value for cycle register TGRA_3 of H'FFFF, when a TGRA_3 compare-match has been specified without synchronous setting for the counter clear source.

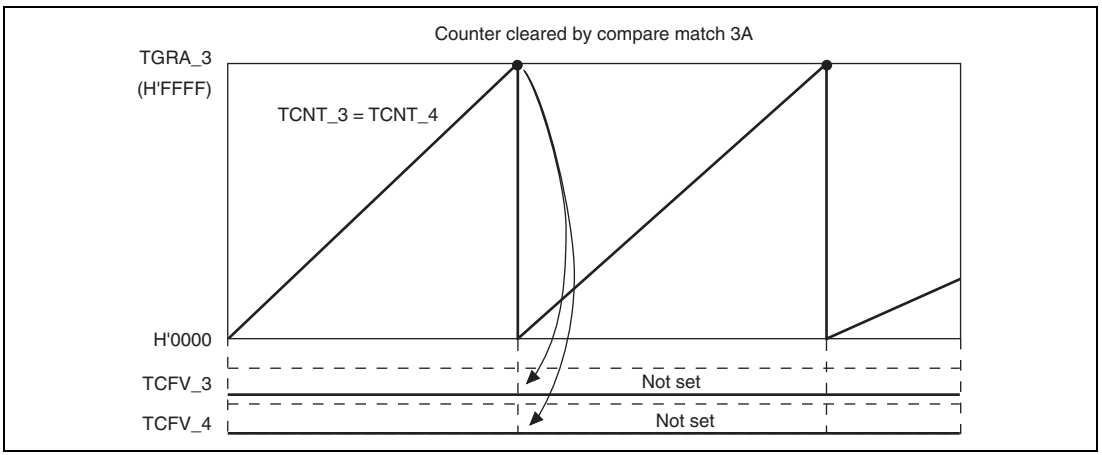


Figure 11.134 Reset Synchronous PWM Mode Overflow Flag

11.7.17 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 11.135 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

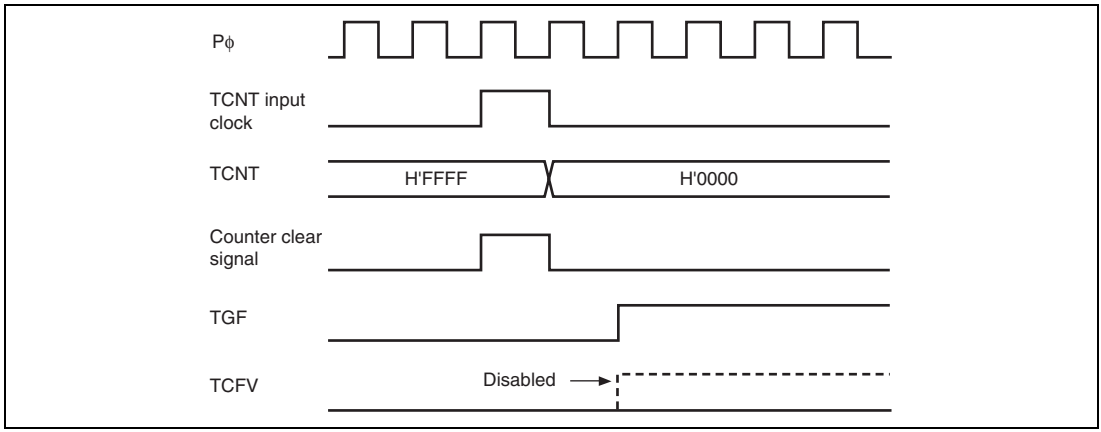


Figure 11.135 Contention between Overflow and Counter Clearing

11.7.18 Contention between TCNT Write and Overflow/Underflow

If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 11.136 shows the operation timing when there is contention between TCNT write and overflow.

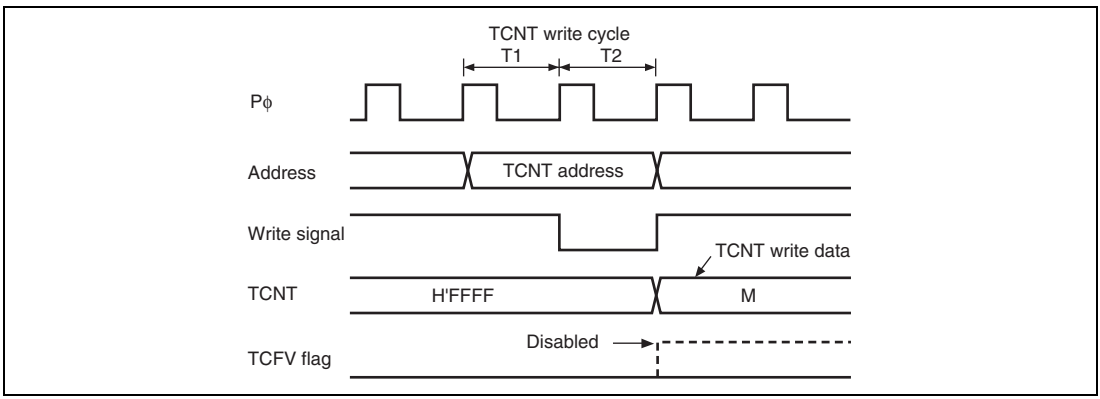


Figure 11.136 Contention between TCNT Write and Overflow

11.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from channel 3 or 4 normal operation or PWM mode 1 to reset-synchronized PWM mode, if the counter is halted with the output pins (TIOC3B, TIOC3D, TIOC4A, TIOC4C, TIOC4B, TIOC4D) in the high-level state, followed by the transition to reset-synchronized PWM mode and operation in that mode, the initial pin output will not be correct.

When making a transition from normal operation to reset-synchronized PWM mode, write H'11 to registers TIORH_3, TIORL_3, TIORH_4, and TIORL_4 to initialize the output pins to low level output, then set an initial register value of H'00 before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, first switch to normal operation, then initialize the output pins to low level output and set an initial register value of H'00 before making the transition to reset-synchronized PWM mode.

11.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When channels 3 and 4 are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is set with the OLSP and OLSN bits in the timer output control register (TOCR). In the case of complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to H'00.

11.7.21 Interrupts in Module Standby Mode

If module standby mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC activation source. Interrupts should therefore be disabled before entering module standby mode.

11.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection

When timer counters 1 and 2 (TCNT_1 and TCNT_2) are operated as a 32-bit counter in cascade connection, the cascade counter value cannot be captured successfully even if input-capture input is simultaneously done to TIOC1A and TIOC2A or to TIOC1B and TIOC2B. This is because the input timing of TIOC1A and TIOC2A or of TIOC1B and TIOC2B may not be the same when external input-capture signals to be input into TCNT_1 and TCNT_2 are taken in synchronization with the internal clock. For example, TCNT_1 (the counter for upper 16 bits) does not capture the count-up value by overflow from TCNT_2 (the counter for lower 16 bits) but captures the count value before the count-up. In this case, the values of TCNT_1 = H'FFF1 and TCNT_2 = H'0000 should be transferred to TGRA_1 and TGRA_2 or to TGRB_1 and TGRB_2, but the values of TCNT_1 = H'FFF0 and TCNT_2 = H'0000 are erroneously transferred.

The MTU2 has a new function that allows simultaneous capture of TCNT_1 and TCNT_2 with a single input-capture as the trigger. This function allows reading of the 32-bit counter such that TCNT_1 and TCNT_2 are captured at the same time. For details, see section 11.3.8, Timer Input Capture Control Register (TICCR).

11.7.23 Note on Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

If either condition (1) or (2) is satisfied when output waveform control at synchronous counter clearing is enabled (WRE bit in TWCR is 1) in complementary PWM mode, the following phenomena occur.

- The dead time of the PWM output pins becomes shorter (or disappears).
- An active level is output from a PWM reverse phase output pin during a period other than the active level output period.

Condition (1) In the initial output suppression period (10), synchronous clearing is performed while the PWM output is in the dead time (figure 11.137).

Condition (2) In the initial output suppression periods (10) and (11), synchronous clearing is performed while TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, or TGRB_4 ≤ TDDR is satisfied (figure 11.138).

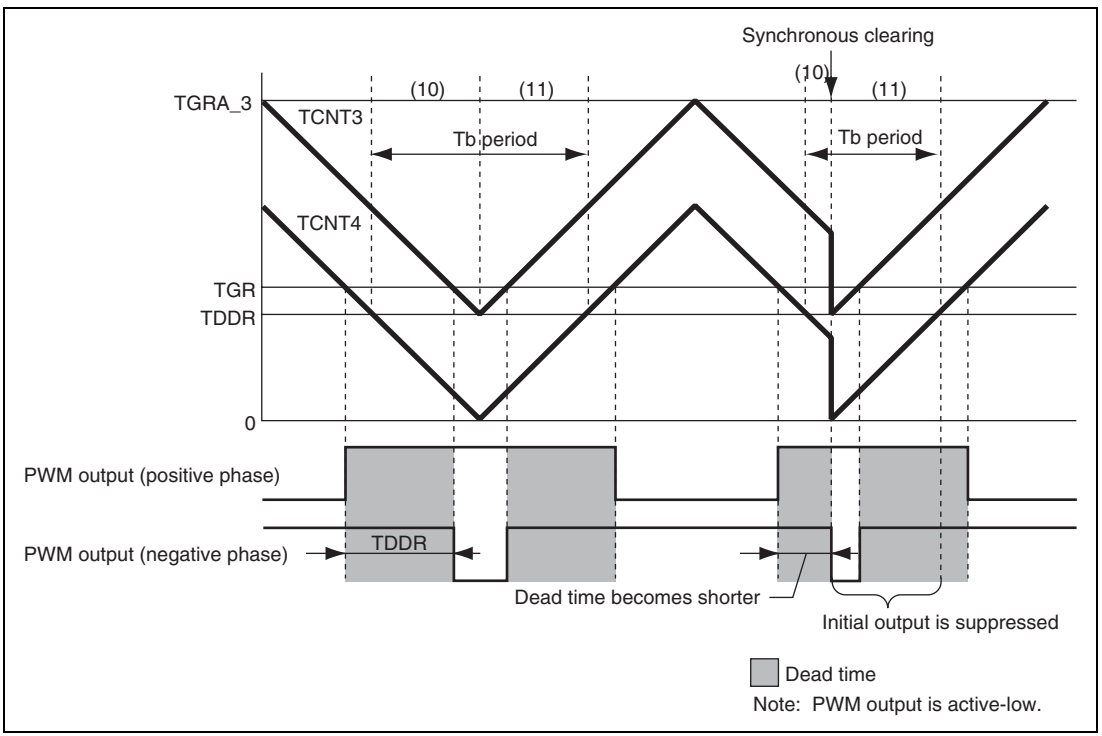


Figure 11.137 Example of Synchronous Clearing under Condition (1)

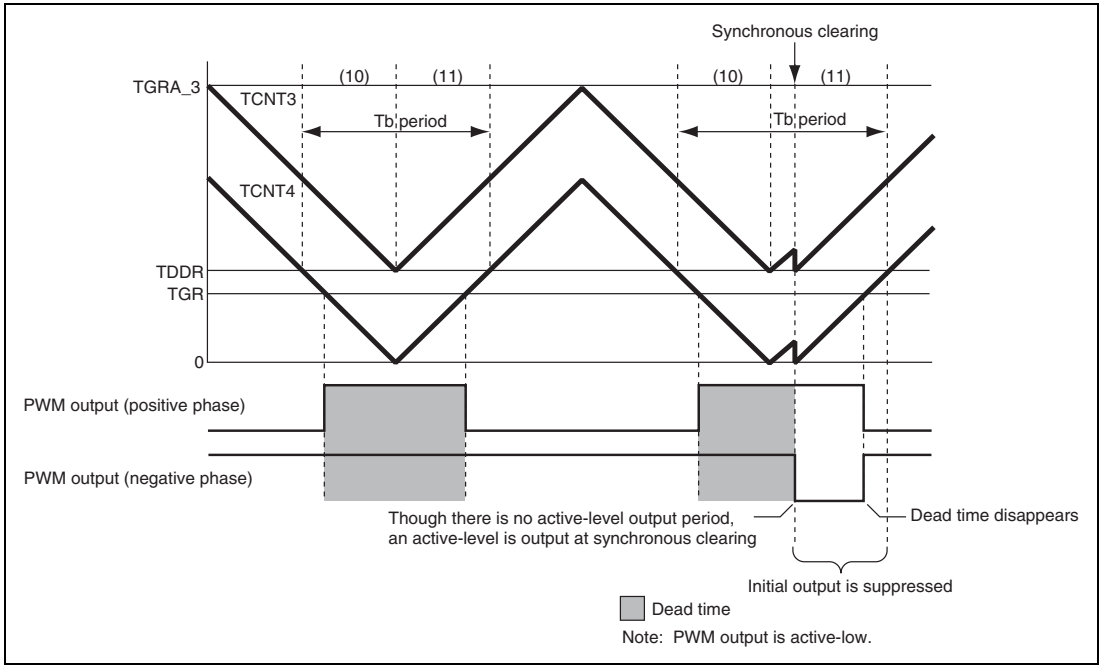


Figure 11.138 Example of Synchronous Clearing under Condition (2)

The above phenomena can be avoided by the following method.

Perform synchronous clearing after compare registers TGRB_3, TGRA_4, and TGRB_4 are all set to be at least twice of the setting of the timer dead time data register (TDDR).

11.8 MTU2 Output Pin Initialization

11.8.1 Operating Modes

The MTU2 has the following six operating modes. Waveform output is possible in all of these modes.

- Normal mode (channels 0 to 4)
- PWM mode 1 (channels 0 to 4)
- PWM mode 2 (channels 0 to 2)
- Phase counting modes 1 to 4 (channels 1 and 2)
- Complementary PWM mode (channels 3 and 4)
- Reset-synchronized PWM mode (channels 3 and 4)

The MTU2 output pin initialization method for each of these modes is described in this section.

11.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a reset and in standby mode. Since MTU2 pin function selection is performed by the pin function controller (PFC), when the PFC is set, the MTU2 pin states at that point are output to the ports. When MTU2 output is selected by the PFC immediately after a reset, the MTU2 output initial level, low, is output directly at the port. When the active level is low, the system will operate at this point, and therefore the PFC setting should be made after initialization of the MTU2 output pins is completed.

Note: Channel number and port notation are substituted for *.

11.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc.

If an error occurs during MTU2 operation, MTU2 output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. For large-current 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, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU2 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. Possible mode transition combinations are shown in table 11.59.

Table 11.59 Mode Transition Combinations

	After					
Before	**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)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23) (24)	(25)
RPWM	(26)	(27)	None	None	(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

11.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc.

- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, 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 pin, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM, RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode and perform initialization with TIOR, then restore TIOR to its initial value, and temporarily disable channel 3 and 4 output with the timer output master enable register (TOER). Then operate the unit in accordance with the mode setting procedure (TOCR setting, TMDR setting, TOER setting).

Note: Channel number is substituted for * indicated in this article.

Pin initialization procedures are described below for the numbered combinations in table 11.59. The active level is assumed to be low.

(1) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.139 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

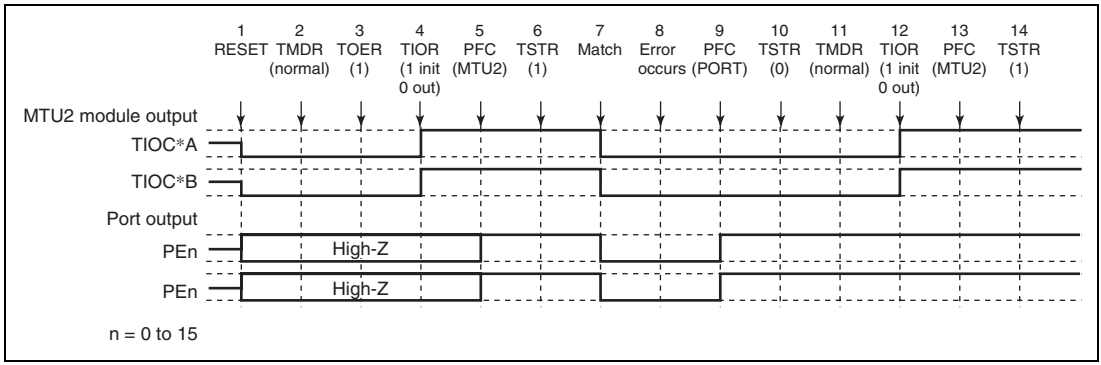


Figure 11.139 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. After a reset, the TMDR setting is for normal mode.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Not necessary when restarting in normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(2) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 11.140 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

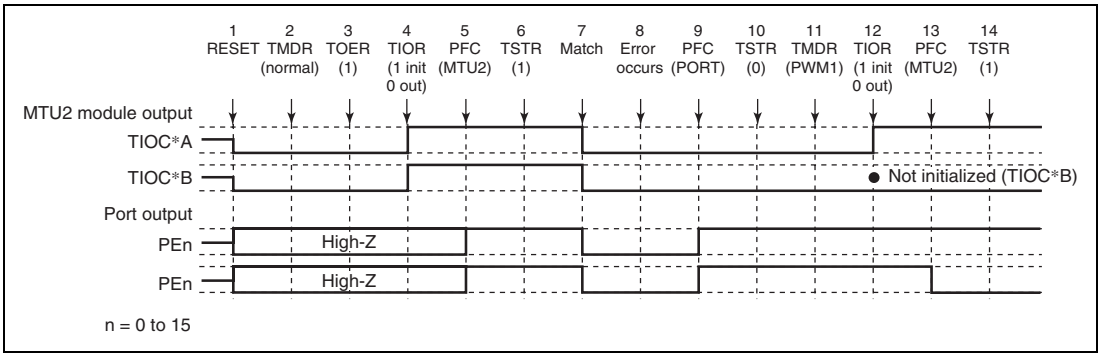


Figure 11.140 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.139.

11. Set PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 1.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(3) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 11.141 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

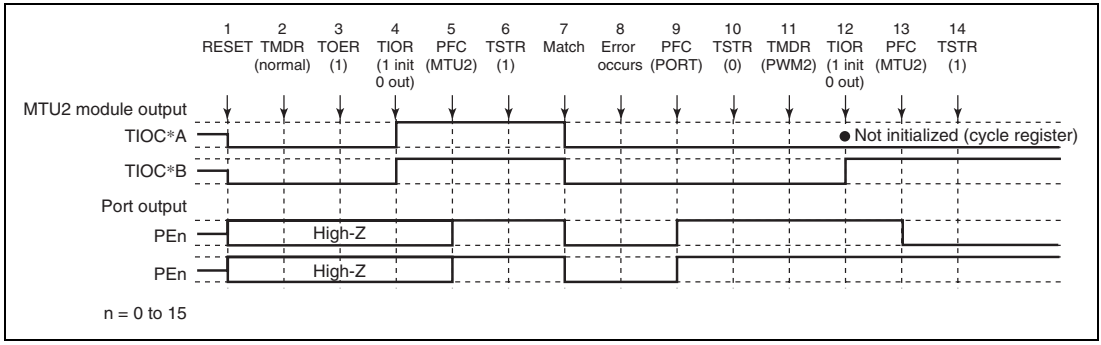


Figure 11.141 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.139.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 11.142 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

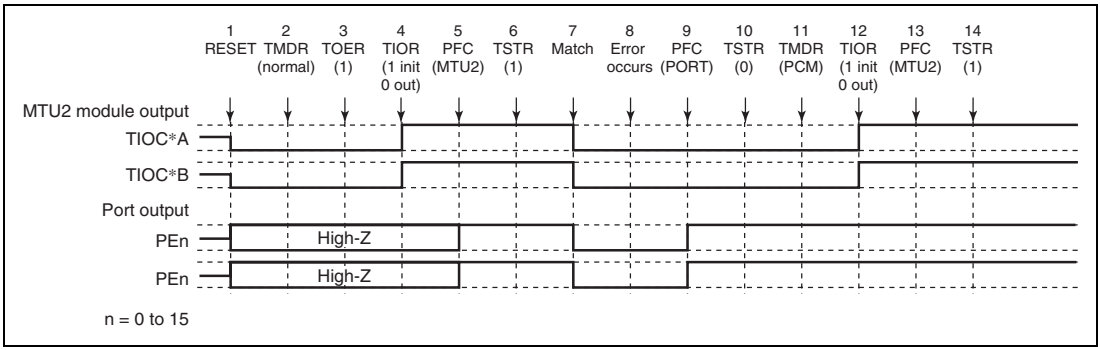


Figure 11.142 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.139.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(5) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 11.143 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

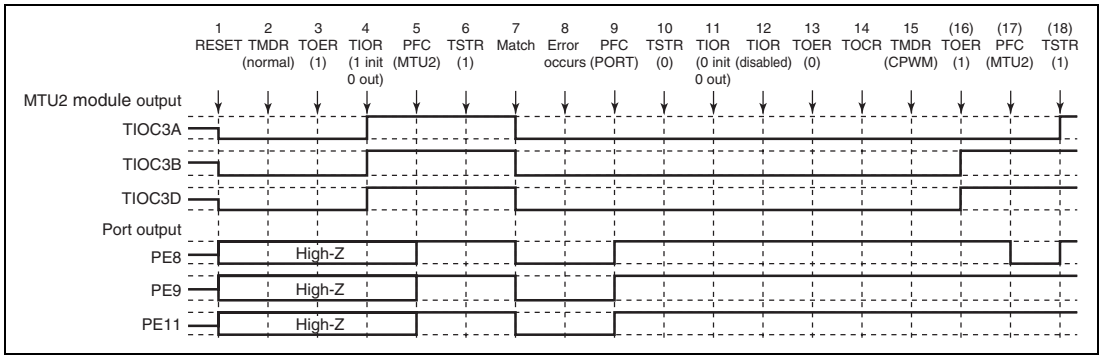


Figure 11.143 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.139.

11. Initialize the normal mode waveform generation section with TIOR.
12. Disable operation of the normal mode waveform generation section with TIOR.
13. Disable channel 3 and 4 output with TOER.
14. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
15. Set complementary PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 output with the PFC.
18. Operation is restarted by TSTR.

(6) Operation when Error Occurs during Normal Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 11.144 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

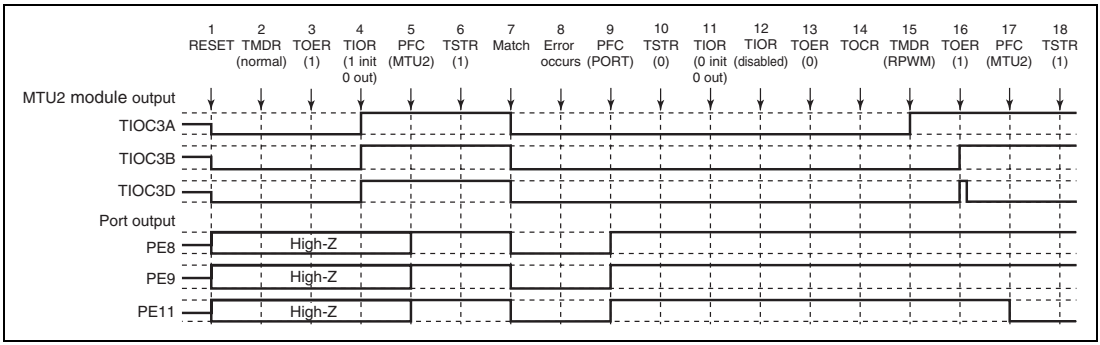


Figure 11.144 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 11.139.

14. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
15. Set reset-synchronized PWM.
16. Enable channel 3 and 4 output with TOER.
17. Set MTU2 output with the PFC.
18. Operation is restarted by TSTR.

(7) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Normal Mode

Figure 11.145 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

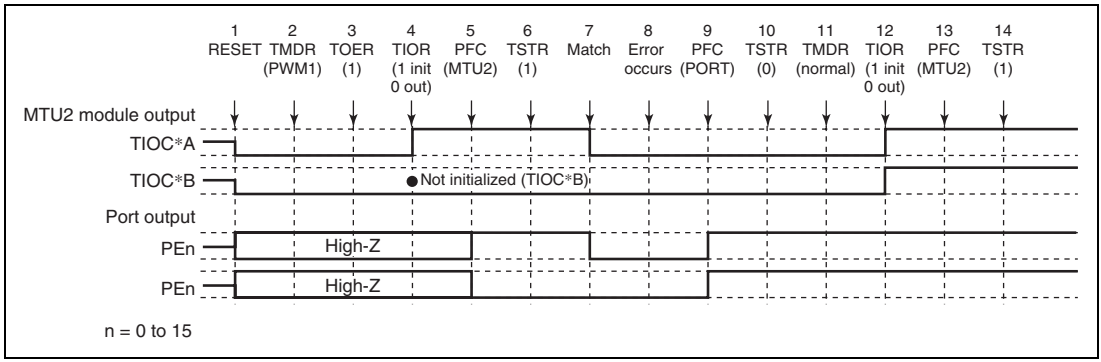


Figure 11.145 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 1.
3. For channels 3 and 4, enable output with TOER before initializing the pins with TIOR.
4. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 1, the TIOC*B side is not initialized.)
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. Output goes low on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR.
11. Set normal mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(8) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 1

Figure 11.146 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

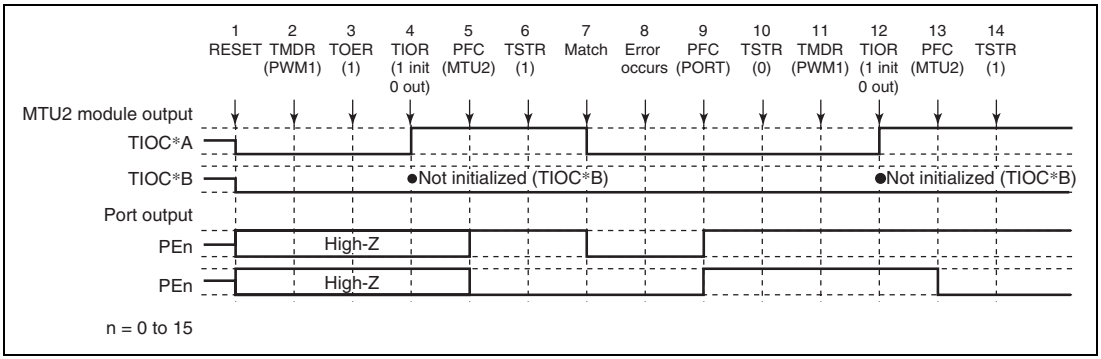


Figure 11.146 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.145.

11. Not necessary when restarting in PWM mode 1.
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(9) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in PWM Mode 2

Figure 11.147 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

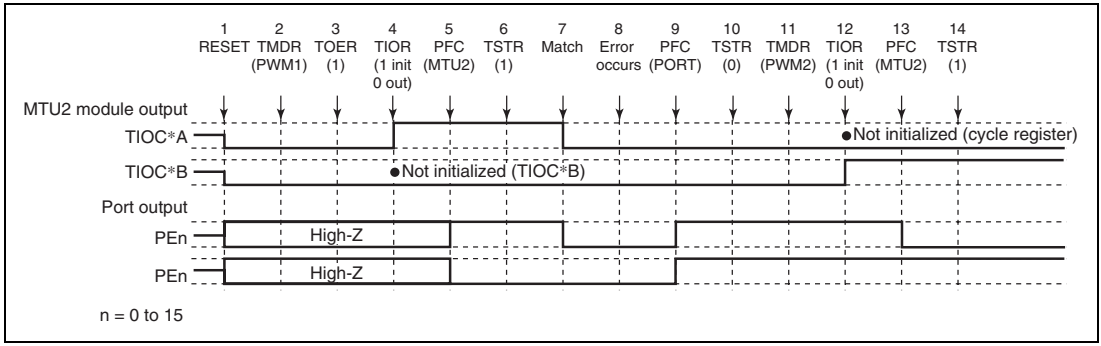


Figure 11.147 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.145.

11. Set PWM mode 2.
12. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: PWM mode 2 can only be set for channels 0 to 2, and therefore TOER setting is not necessary.

(10) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Phase Counting Mode

Figure 11.148 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

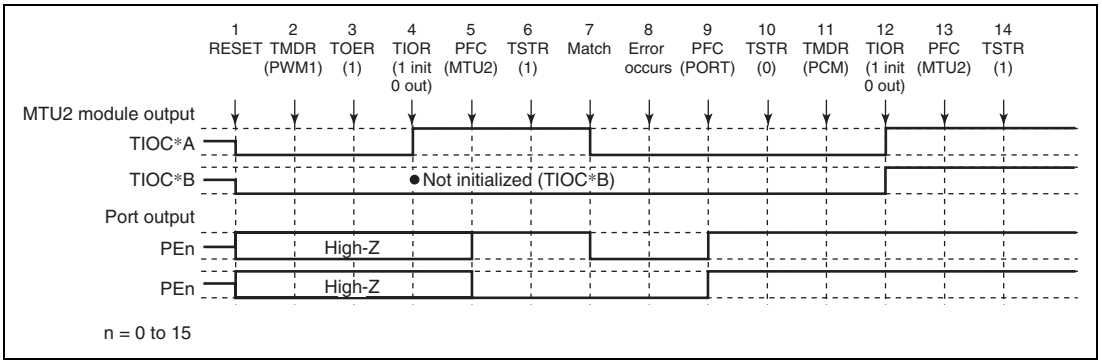


Figure 11.148 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.145.

11. Set phase counting mode.
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

Note: Phase counting mode can only be set for channels 1 and 2, and therefore TOER setting is not necessary.

(11) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Complementary PWM Mode

Figure 11.149 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

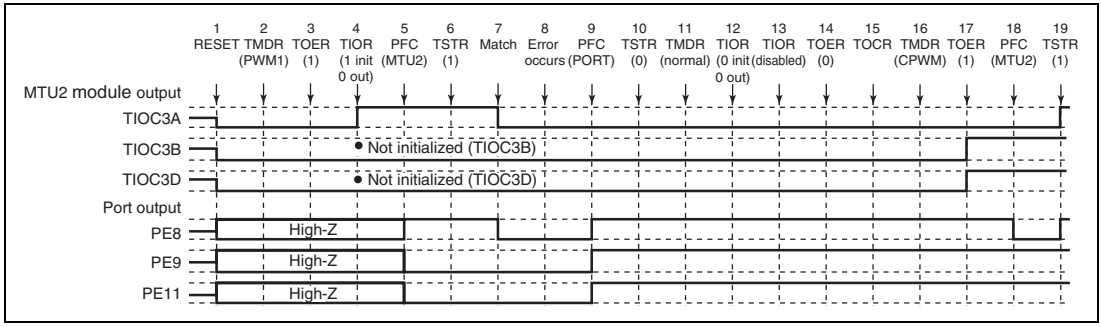


Figure 11.149 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.145.

11. Set normal mode for initialization of the normal mode waveform generation section.
12. Initialize the PWM mode 1 waveform generation section with TIOR.
13. Disable operation of the PWM mode 1 waveform generation section with TIOR.
14. Disable channel 3 and 4 output with TOER.
15. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
16. Set complementary PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 output with the PFC.
19. Operation is restarted by TSTR.

(12) Operation when Error Occurs during PWM Mode 1 Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 11.150 shows an explanatory diagram of the case where an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

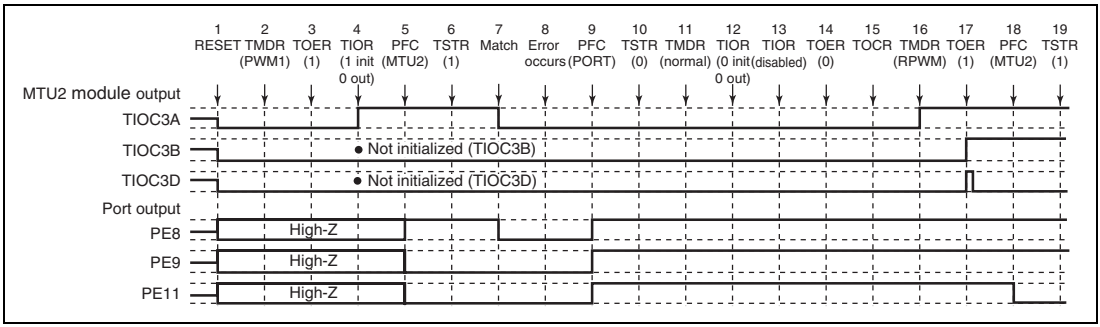


Figure 11.150 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 11.149.

15. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
16. Set reset-synchronized PWM.
17. Enable channel 3 and 4 output with TOER.
18. Set MTU2 output with the PFC.
19. Operation is restarted by TSTR.

(13) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Normal Mode

Figure 11.151 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

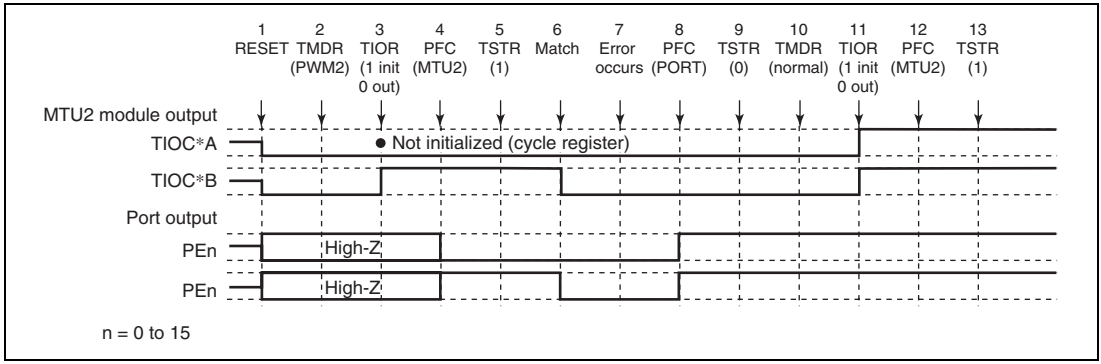


Figure 11.151 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set PWM mode 2.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, TIOC *A is the cycle register.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(14) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 1

Figure 11.152 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

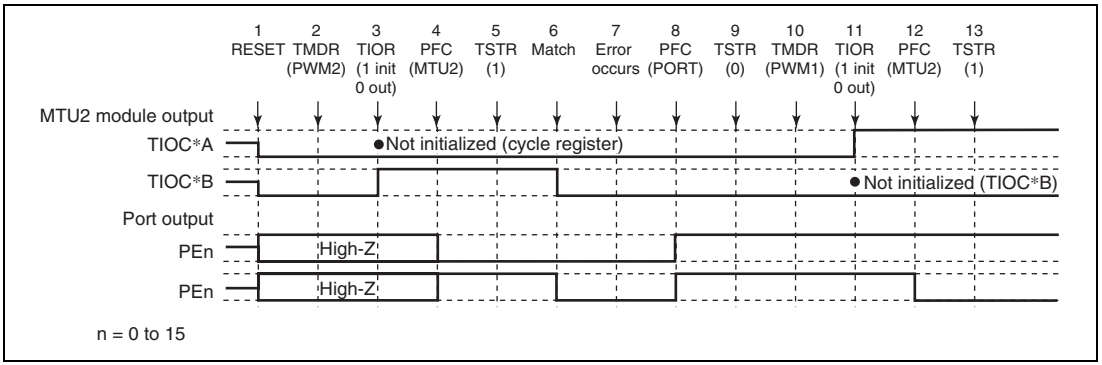


Figure 11.152 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.151.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(15) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2

Figure 11.153 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

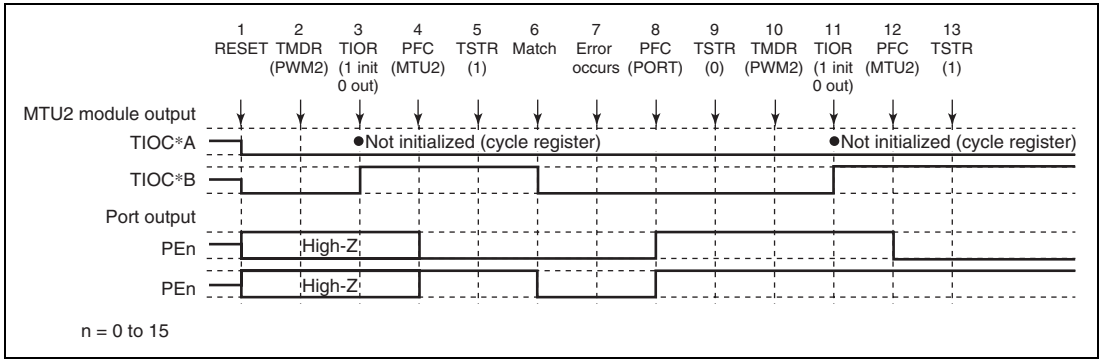


Figure 11.153 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.151.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(16) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in Phase Counting Mode

Figure 11.154 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

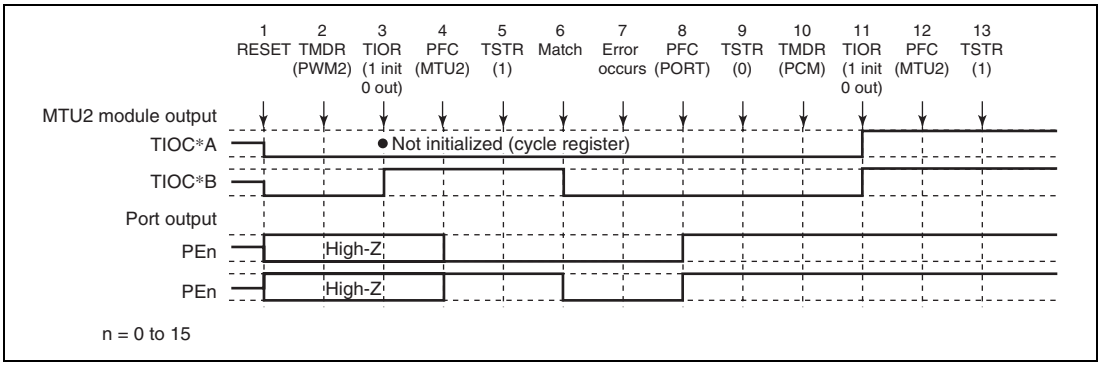


Figure 11.154 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.151.

10. Set phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(17) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.155 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

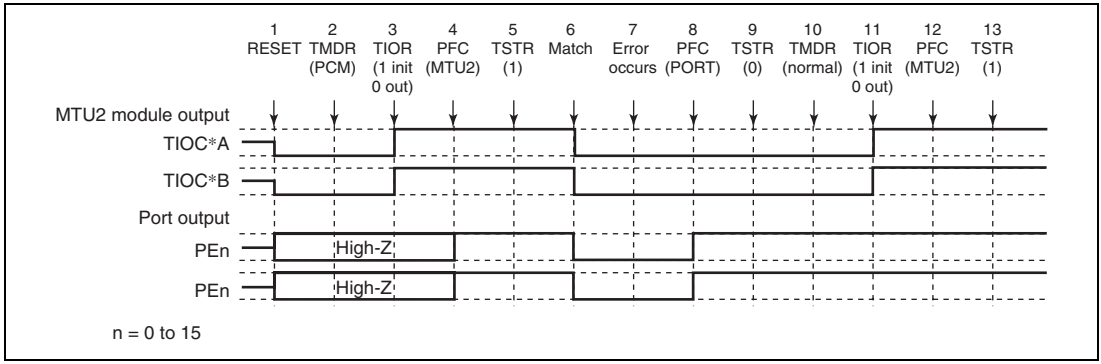


Figure 11.155 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Set phase counting mode.
3. Initialize the pins with TIOR. (The example shows initial high output, with low output on compare-match occurrence.)
4. Set MTU2 output with the PFC.
5. The count operation is started by TSTR.
6. Output goes low on compare-match occurrence.
7. An error occurs.
8. Set port output with the PFC and output the inverse of the active level.
9. The count operation is stopped by TSTR.
10. Set in normal mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(18) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 11.156 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

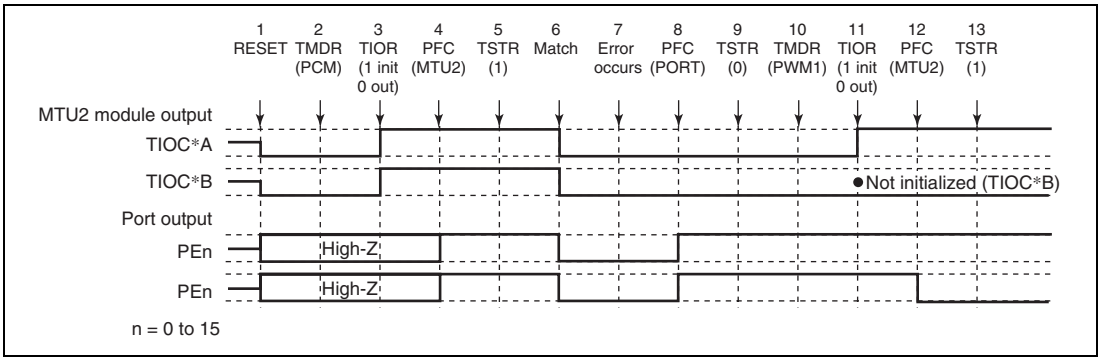


Figure 11.156 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.155.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(19) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 11.157 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

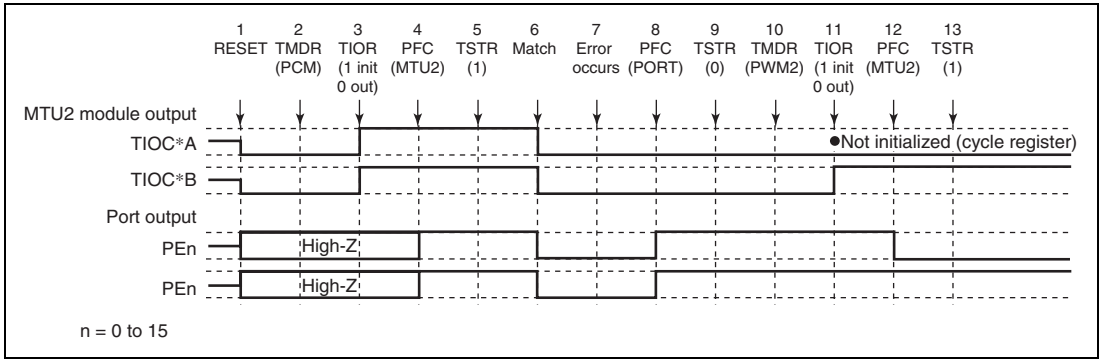


Figure 11.157 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.155.

10. Set PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(20) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in Phase Counting Mode

Figure 11.158 shows an explanatory diagram of the case where an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

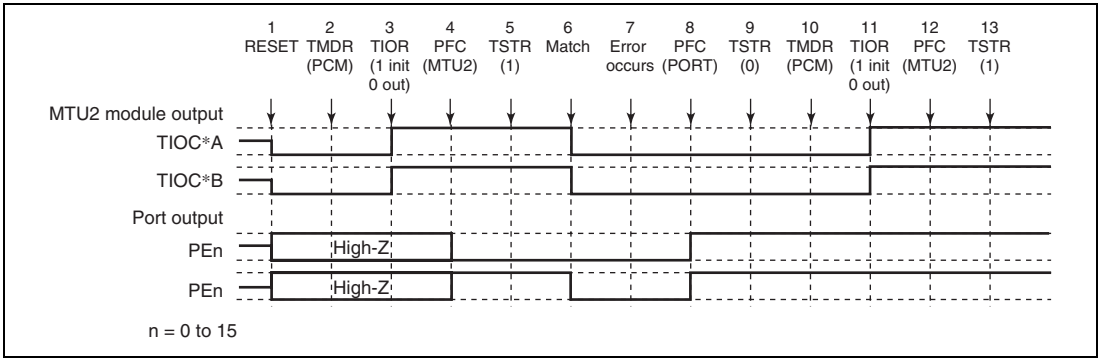


Figure 11.158 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.155.

10. Not necessary when restarting in phase counting mode.
11. Initialize the pins with TIOR.
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(21) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.159 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

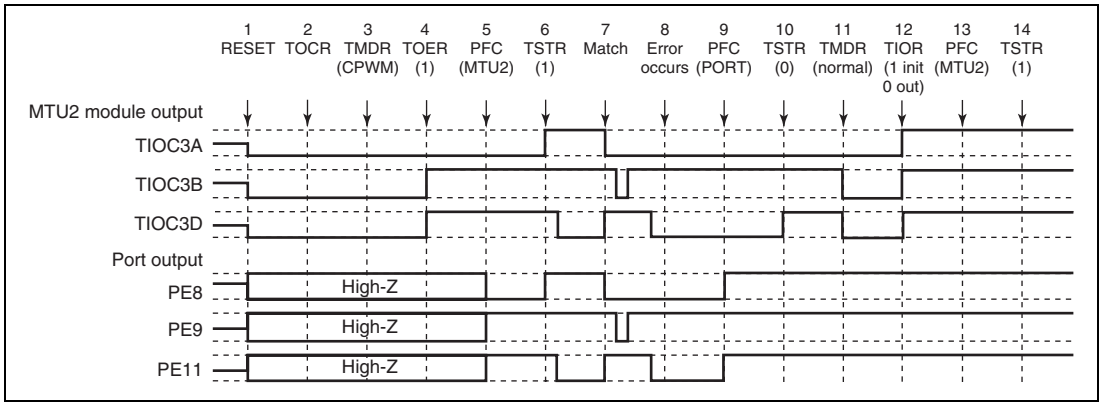


Figure 11.159 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
3. Set complementary PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The complementary PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the complementary PWM output initial value.)
11. Set normal mode. (MTU2 output goes low.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(22) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 11.160 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

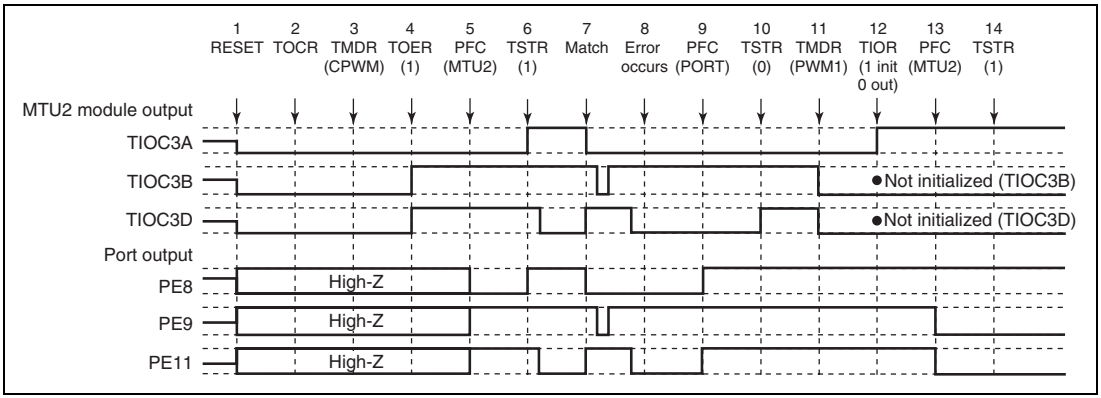


Figure 11.160 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.159.

11. Set PWM mode 1. (MTU2 output goes low.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(23) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 11.161 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the cycle and duty settings at the time the counter was stopped).

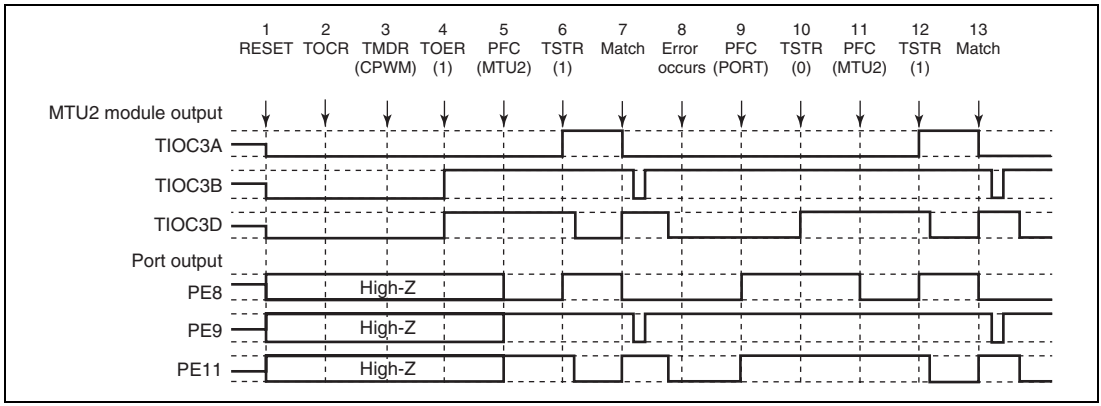


Figure 11.161 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.159.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The complementary PWM waveform is output on compare-match occurrence.

(24) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 11.162 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using completely new cycle and duty settings).

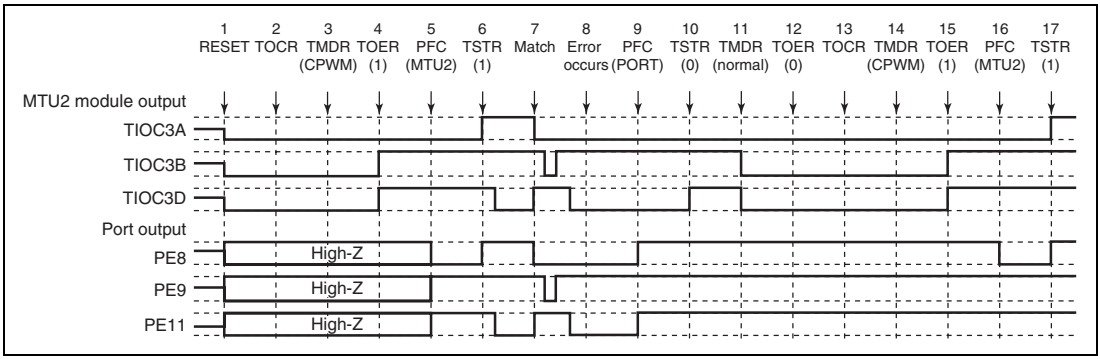


Figure 11.162 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.159.

11. Set normal mode and make new settings. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the complementary PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set complementary PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

(25) Operation when Error Occurs during Complementary PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 11.163 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

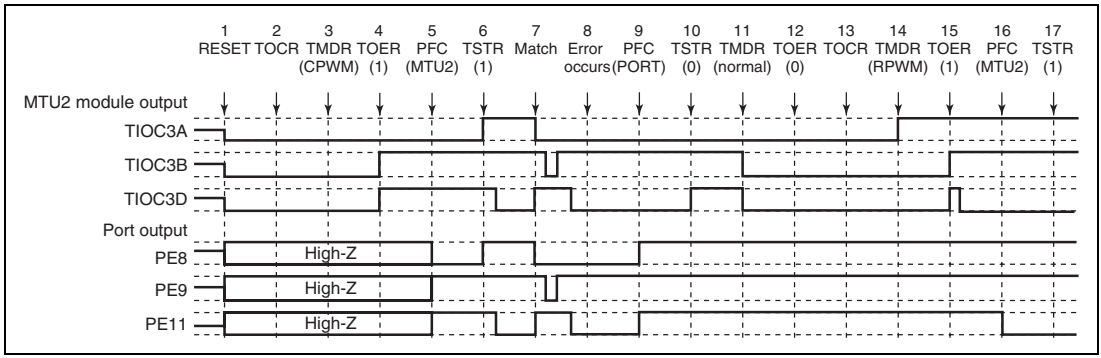


Figure 11.163 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.159.

11. Set normal mode. (MTU2 output goes low.)
12. Disable channel 3 and 4 output with TOER.
13. Select the reset-synchronized PWM mode output level and cyclic output enabling/disabling with TOCR.
14. Set reset-synchronized PWM.
15. Enable channel 3 and 4 output with TOER.
16. Set MTU2 output with the PFC.
17. Operation is restarted by TSTR.

(26) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Normal Mode

Figure 11.164 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

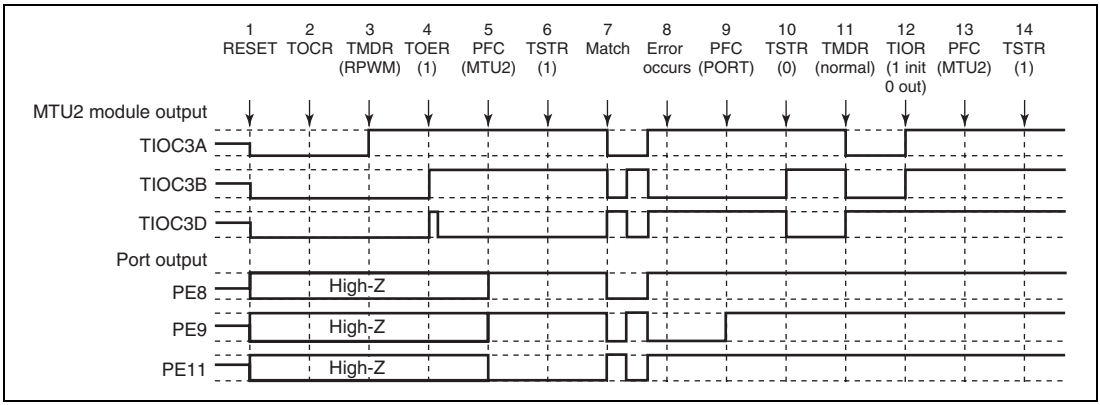


Figure 11.164 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

1. After a reset, MTU2 output is low and ports are in the high-impedance state.
2. Select the reset-synchronized PWM output level and cyclic output enabling/disabling with TOCR.
3. Set reset-synchronized PWM.
4. Enable channel 3 and 4 output with TOER.
5. Set MTU2 output with the PFC.
6. The count operation is started by TSTR.
7. The reset-synchronized PWM waveform is output on compare-match occurrence.
8. An error occurs.
9. Set port output with the PFC and output the inverse of the active level.
10. The count operation is stopped by TSTR. (MTU2 output becomes the reset-synchronized PWM output initial value.)
11. Set normal mode. (MTU2 positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR.
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(27) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in PWM Mode 1

Figure 11.165 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

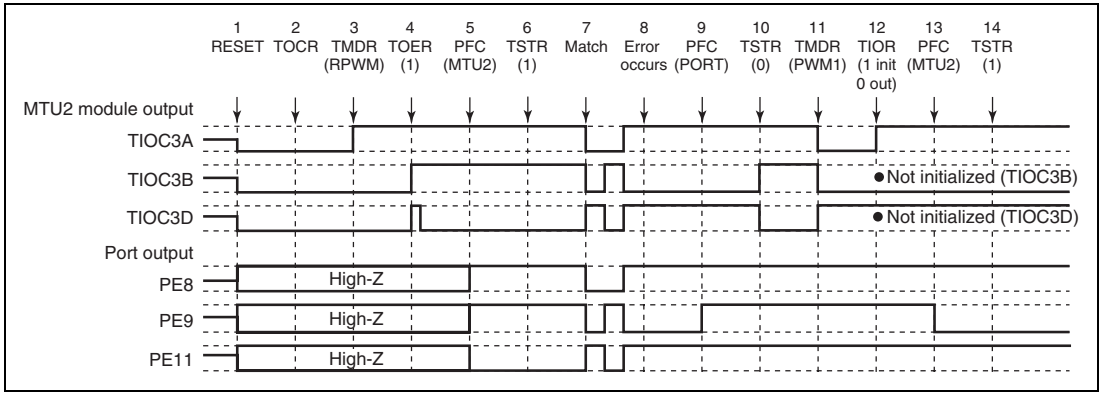


Figure 11.165 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.164.

11. Set PWM mode 1. (MTU2 positive phase output is low, and negative phase output is high.)
12. Initialize the pins with TIOR. (In PWM mode 1, the TIOC *B side is not initialized.)
13. Set MTU2 output with the PFC.
14. Operation is restarted by TSTR.

(28) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Complementary PWM Mode

Figure 11.166 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

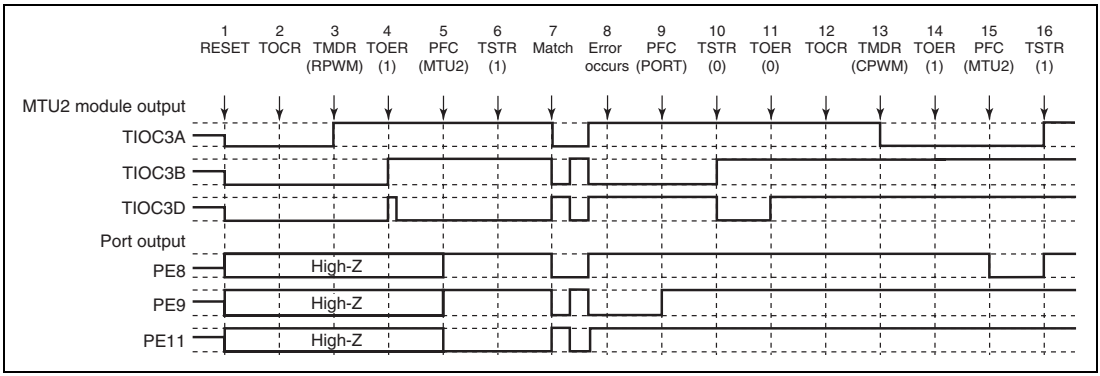


Figure 11.166 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.164.

11. Disable channel 3 and 4 output with TOER.
12. Select the complementary PWM output level and cyclic output enabling/disabling with TOCR.
13. Set complementary PWM. (The MTU2 cyclic output pin goes low.)
14. Enable channel 3 and 4 output with TOER.
15. Set MTU2 output with the PFC.
16. Operation is restarted by TSTR.

(29) Operation when Error Occurs during Reset-Synchronized PWM Mode Operation, and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 11.167 shows an explanatory diagram of the case where an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

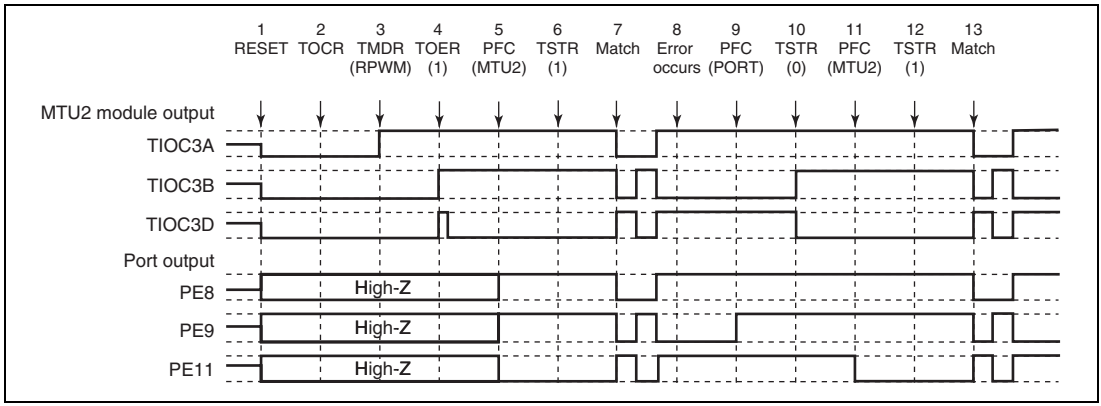


Figure 11.167 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.164.

11. Set MTU2 output with the PFC.
12. Operation is restarted by TSTR.
13. The reset-synchronized PWM waveform is output on compare-match occurrence.

Section 12 Multi-Function Timer Pulse Unit 2S (MTU2S)

This LSI has an on-chip multi-function timer pulse unit 2S (MTU2S) that comprises three 16-bit timer channels. The MTU2S includes channels 3 to 5 of the MTU2. For details, refer to section 11, Multi-Function Timer Pulse Unit 2 (MTU2). To distinguish from the MTU2, "S" is added to the end of the MTU2S input/output pin and register names. For example, TIOC3A is called TIOC3AS and TGRA_3 is called TGRA_3S in this section.

The MTU2S can operate at 100 MHz max. for complementary PWM output functions or at 50 MHz max. for the other functions.

Table 12.1 MTU2S Functions

Item		Channel 3	Channel 4	Channel 5
Count clock		Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 Mϕ/1024	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 Mϕ/1024	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64
General registers		TGRA_3S TGRB_3S	TGRA_4S TGRB_4S	TGRU_5S TGRV_5S TGRW_5S
General registers/ buffer registers		TGRC_3S TGRD_3S	TGRC_4S TGRD_4S	—
I/O pins		TIOC3AS TIOC3BS TIOC3CS TIOC3DS	TIOC4AS TIOC4BS TIOC4CS TIOC4DS	Input pins TIC5US TIC5VS TIC5WS
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	√	√	—
	1 output	√	√	—
	Toggle output	√	√	—
Input capture function		√	√	√
Synchronous operation		√	√	—

Item	Channel 3	Channel 4	Channel 5
PWM mode 1	√	√	—
PWM mode 2	—	—	—
Complementary PWM mode	√	√	—
Reset PWM mode	√	√	—
AC synchronous motor drive mode	—	—	—
Phase counting mode	—	—	—
Buffer operation	√	√	—
Counter function of compensation for dead time	—	—	√
DTC activation	TGR compare match or input capture	TGR compare match or input capture, or TCNT overflow or underflow	TGR compare match or input capture
A/D converter start trigger	TGRA_3S compare match or input capture	TGRA_4S compare match or input capture TCNT_4S underflow (trough) in complementary PWM mode	—
Interrupt sources	5 sources • Compare match or input capture 3AS • Compare match or input capture 3BS • Compare match or input capture 3CS • Compare match or input capture 3DS • Overflow	5 sources • Compare match or input capture 4AS • Compare match or input capture 4BS • Compare match or input capture 4CS • Compare match or input capture 4DS • Overflow or underflow	3 sources • Compare match or input capture 5US • Compare match or input capture 5VS • Compare match or input capture 5WS

Item	Channel 3	Channel 4	Channel 5
A/D converter start request delaying function	—	• A/D converter start request at a match between TADCORA_4S and TCNT_4S • A/D converter start request at a match between TADCORB_4S and TCNT_4S	—
Interrupt skipping function	• Skips TGRA_3S compare match interrupts	• Skips TCIV_4S interrupts	—

[Legend]
√: Possible
—: Not possible

12.1 Input/Output Pins

Table 12.2 Pin Configuration

Channel	Symbol	I/O	Function
3	TIOC3AS	I/O	TGRA_3S input capture input/output compare output/PWM output pin
	TIOC3BS	I/O	TGRB_3S input capture input/output compare output/PWM output pin
	TIOC3CS	I/O	TGRC_3S input capture input/output compare output/PWM output pin
	TIOC3DS	I/O	TGRD_3S input capture input/output compare output/PWM output pin
4	TIOC4AS	I/O	TGRA_4S input capture input/output compare output/PWM output pin
	TIOC4BS	I/O	TGRB_4S input capture input/output compare output/PWM output pin
	TIOC4CS	I/O	TGRC_4S input capture input/output compare output/PWM output pin
	TIOC4DS	I/O	TGRD_4S input capture input/output compare output/PWM output pin
5	TIC5US	Input	TGRU_5S input capture input/external pulse input pin
	TIC5VS	Input	TGRV_5S input capture input/external pulse input pin
	TIC5WS	Input	TGRW_5S input capture input/external pulse input pin

12.2 Register Descriptions

The MTU2S has the following registers. For details on register addresses and register states during each process, refer to section 32, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 3 is expressed as TCR_3S.

Table 12.3 Register Configuration

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer control register_3S	TCR_3S	R/W	H'00	H'FFFE4A00	8, 16, 32
Timer control register_4S	TCR_4S	R/W	H'00	H'FFFE4A01	8
Timer mode register_3S	TMDR_3S	R/W	H'00	H'FFFE4A02	8, 16
Timer mode register_4S	TMDR_4S	R/W	H'00	H'FFFE4A03	8
Timer I/O control register H_3S	TIORH_3S	R/W	H'00	H'FFFE4A04	8, 16, 32
Timer I/O control register L_3S	TIORL_3S	R/W	H'00	H'FFFE4A05	8
Timer I/O control register H_4S	TIORH_4S	R/W	H'00	H'FFFE4A06	8, 16
Timer I/O control register L_4S	TIORL_4S	R/W	H'00	H'FFFE4A07	8
Timer interrupt enable register_3S	TIER_3S	R/W	H'00	H'FFFE4A08	8, 16
Timer interrupt enable register_4S	TIER_4S	R/W	H'00	H'FFFE4A09	8
Timer output master enable register S	TOERS	R/W	H'C0	H'FFFE4A0A	8
Timer gate control register S	TGCRS	R/W	H'80	H'FFFE4A0D	8
Timer output control register 1S	TOCR1S	R/W	H'00	H'FFFE4A0E	8, 16
Timer output control register 2S	TOCR2S	R/W	H'00	H'FFFE4A0F	8
Timer counter_3S	TCNT_3S	R/W	H'0000	H'FFFE4A10	16, 32
Timer counter_4S	TCNT_4S	R/W	H'0000	H'FFFE4A12	16
Timer cycle data register S	TCDRS	R/W	H'FFFF	H'FFFE4A14	16, 32
Timer dead time data register S	TDDRS	R/W	H'FFFF	H'FFFE4A16	16
Timer general register A_3S	TGRA_3S	R/W	H'FFFF	H'FFFE4A18	16, 32
Timer general register B_3S	TGRB_3S	R/W	H'FFFF	H'FFFE4A1A	16
Timer general register A_4S	TGRA_4S	R/W	H'FFFF	H'FFFE4A1C	16, 32
Timer general register B_4S	TGRB_4S	R/W	H'FFFF	H'FFFE4A1E	16
Timer subcounter S	TCNTSS	R	H'0000	H'FFFE4A20	16, 32
Timer cycle buffer register S	TCBRS	R/W	H'FFFF	H'FFFE4A22	16

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer general register C_3S	TGRC_3S	R/W	H'FFFF	H'FFFE4A24	16, 32
Timer general register D_3S	TGRD_3S	R/W	H'FFFF	H'FFFE4A26	16
Timer general register C_4S	TGRC_4S	R/W	H'FFFF	H'FFFE4A28	16, 32
Timer general register D_4S	TGRD_4S	R/W	H'FFFF	H'FFFE4A2A	16
Timer status register_3S	TSR_3S	R/W	H'C0	H'FFFE4A2C	8, 16
Timer status register_4S	TSR_4S	R/W	H'C0	H'FFFE4A2D	8
Timer interrupt skipping set register S	TITCRS	R/W	H'00	H'FFFE4A30	8, 16
Timer interrupt skipping counter S	TITCNTS	R	H'00	H'FFFE4A31	8
Timer buffer transfer set register S	TBTERS	R/W	H'00	H'FFFE4A32	8
Timer dead time enable register S	TDERS	R/W	H'01	H'FFFE4A34	8
Timer output level buffer register S	TOLBRS	R/W	H'00	H'FFFE4A36	8
Timer buffer operation transfer mode register_3S	TBTM_3S	R/W	H'00	H'FFFE4A38	8, 16
Timer buffer operation transfer mode register_4S	TBTM_4S	R/W	H'00	H'FFFE4A39	8
Timer A/D converter start request control register S	TADCRS	R/W	H'0000	H'FFFE4A40	16
Timer A/D converter start request cycle set register A_4S	TADCORA_4S	R/W	H'FFFF	H'FFFE4A44	16, 32
Timer A/D converter start request cycle set register B_4S	TADCORB_4S	R/W	H'FFFF	H'FFFE4A46	16
Timer A/D converter start request cycle set buffer register A_4S	TADCOBRA_4S	R/W	H'FFFF	H'FFFE4A48	16, 32
Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFE4A4A	16
Timer synchronous clear register S*	TSYCRS	R/W	H'00	H'FFFE4A50	8
Timer waveform control register S	TWCRS	R/W	H'00	H'FFFE4A60	8
Timer start register S	TSTRS	R/W	H'00	H'FFFE4A80	8, 16
Timer synchronous register S	TSYRS	R/W	H'00	H'FFFE4A81	8
Timer read/write enable register S	TRWERS	R/W	H'01	H'FFFE4A84	8

Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Timer counter U_5S	TCNTU_5S	R/W	H'0000	H'FFFE4880	16, 32
Timer general register U_5S	TGRU_5S	R/W	H'FFFF	H'FFFE4882	16
Timer control register U_5S	TCRU_5S	R/W	H'00	H'FFFE4884	8
Timer I/O control register U_5S	TIORU_5S	R/W	H'00	H'FFFE4886	8
Timer counter V_5S	TCNTV_5S	R/W	H'0000	H'FFFE4890	16, 32
Timer general register V_5S	TGRV_5S	R/W	H'FFFF	H'FFFE4892	16
Timer control register V_5S	TCRV_5S	R/W	H'00	H'FFFE4894	8
Timer I/O control register V_5S	TIORV_5S	R/W	H'00	H'FFFE4896	8
Timer counter W_5S	TCNTW_5S	R/W	H'0000	H'FFFE48A0	16, 32
Timer general register W_5S	TGRW_5S	R/W	H'FFFF	H'FFFE48A2	16
Timer control register W_5S	TCRW_5S	R/W	H'00	H'FFFE48A4	8
Timer I/O control register W_5S	TIORW_5S	R/W	H'00	H'FFFE48A6	8
Timer status register_5S	TSR_5S	R/W	H'00	H'FFFE48B0	8
Timer interrupt enable register_5S	TIER_5S	R/W	H'00	H'FFFE48B2	8
Timer start register_5S	TSTR_5S	R/W	H'00	H'FFFE48B4	8
Timer compare match clear register S	TCNTCMPCLRS	R/W	H'00	H'FFFE48B6	8

Note: * For details on the above registers, see section 11.3.9, Timer Synchronous Clear Register S (TSYCRS) and figure 11.85, Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source in section 11, Multi-Function Timer Pulse Unit 2 (MTU2).

Section 13 Port Output Enable 2 (POE2)

The port output enable 2 (POE2) can be used to place the high-current pins (PE9/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B, PE14/TIOC4C, PE15/TIOC4D, PE0/TIOC4AS, PE1/TIOC4BS, PE2/TIOC4CS, PE3/TIOC4DS, PE5/TIOC3BS, PE6/TIOC3DS, PD15/TIOC4DS, PD14/TIOC4CS, PD13/TIOC4BS, PD12/TIOC4AS, PD11/TIOC3DS, PD10/TIOC3BS, PD24/TIOC4DS, PD25/TIOC4CS, PD26/TIOC4BS, PD27/TIOC4AS, PD28/TIOC3DS, and PD29/TIOC3BS) and the pins for channel 0 of the MTU2 (PE0/TIOC0A, PE1/TIOC0B, PE2/TIOC0C, PE3/TIOC0D, PB1/TIOC0A, PB2/TIOC0B, PB3/TIOC0C, and PB4/TIOC0D) in high-impedance state, depending on the change on the $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ input pins and the output status of the high-current pins, or by modifying register settings. It can also simultaneously generate interrupt requests.

13.1 Features

- Each of the $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ input pins can be set for falling edge, Pϕ/8 × 16, Pϕ/16 × 16, or Pϕ/128 × 16 low-level sampling.
- High-current pins and the pins for channel 0 of the MTU2 can be placed in high-impedance state by $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins falling-edge or low-level sampling.
- High-current pins can be placed in high-impedance state when the high-current pin output levels are compared and simultaneous active-level output continues for one cycle or more.
- High-current pins and the pins for channel 0 of the MTU2 can be placed in high-impedance state by modifying the POE2 register settings.
- Interrupts can be generated by input-level sampling or output-level comparison results.

The POE2 has input level detection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in the block diagram of figure 13.1.

Figure 13.1 shows a block diagram of the POE2.

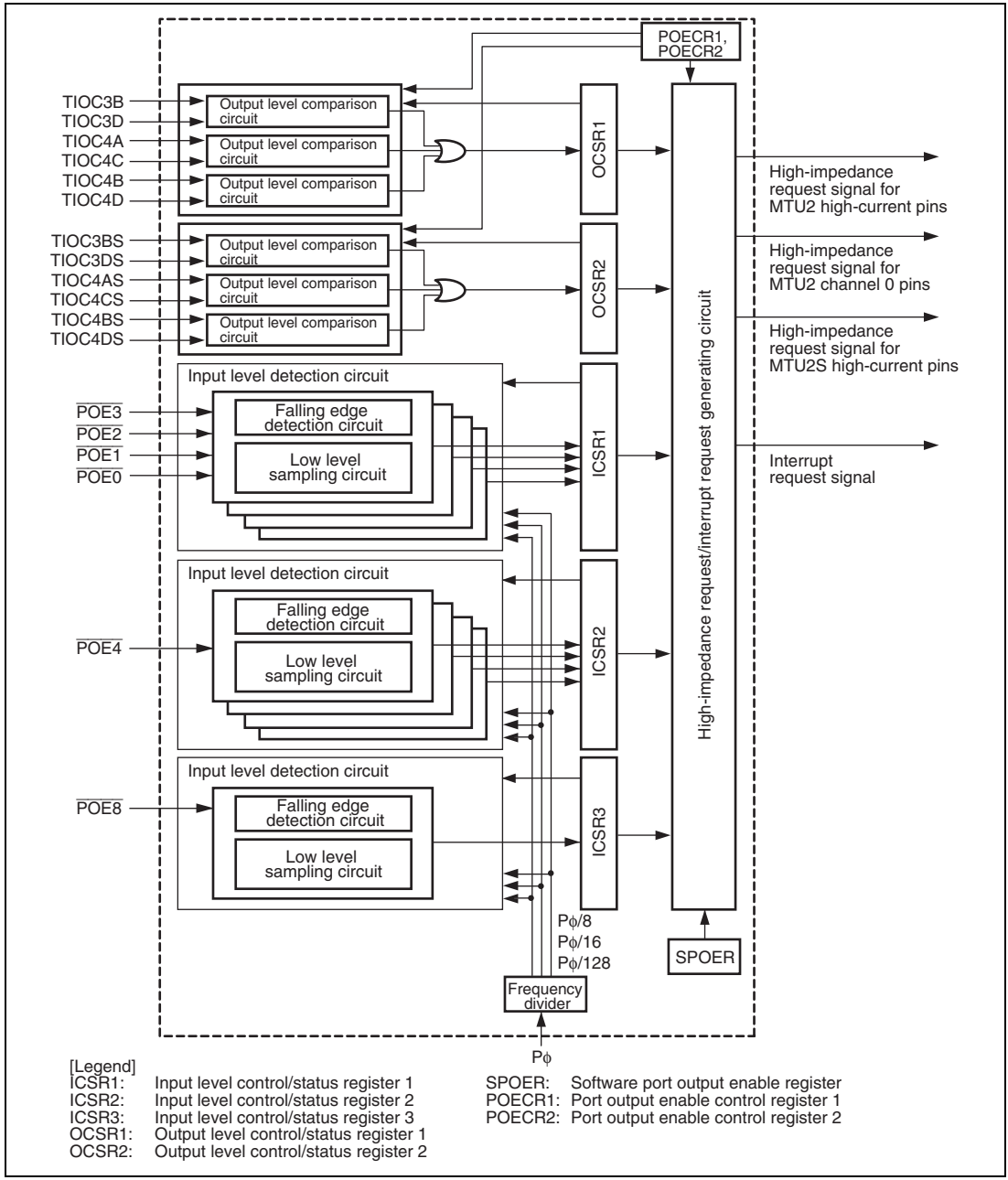


Figure 13.1 Block Diagram of POE2

13.2 Input/Output Pins

Table 13.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Port output enable input pins 0 to 3	$\overline{\text{POE0}}$ to $\overline{\text{POE3}}$	Input	Input request signals to place high-current pins (PE9/TIOC3B, PE11/TIOC3D, PE12/TIOC4A, PE13/TIOC4B, PE14/TIOC4C, and PE15/TIOC4D) for MTU2 in high-impedance state
Port output enable input pins 4 to 7	$\overline{\text{POE4}}$	Input	Input request signals to place high-current pins (PE5/TIOC3BS, PE6/TIOC3DS, PE0/TIOC4AS, PE1/TIOC4BS, PE2/TIOC4CS, PE3/TIOC4DS, PD10/TIOC3BS, PD11/TIOC3DS, PD12/TIOC4AS, PD13/TIOC4BS, PD14/TIOC4CS, PD15/TIOC4DS, PD29/TIOC3BS, PD28/TIOC3DS, PD27/TIOC4AS, PD26/TIOC4BS, PD25/TIOC4CS, and PD24/TIOC4DS) for MTU2S in high-impedance state
Port output enable input pin 8	$\overline{\text{POE8}}$	Input	Inputs a request signal to place pins (PE0/TIOC0A, PE1/TIOC0B, PE2/TIOC0C, PE3/TIOC0D, PB1/TIOC0A, PB2/TIOC0B, PB3/TIOC0C, and PB4/TIOC0D) for channel 0 in MTU2 in high-impedance state

Table 13.2 shows output-level comparisons with pin combinations.

Table 13.2 Pin Combinations

Pin Combination	I/O	Description
PE9/TIOC3B and PE11/TIOC3D PE12/TIOC4A and PE14/TIOC4C PE13/TIOC4B and PE15/TIOC4D	Output	The high-current pins for the MTU2 are placed in high-impedance state when the pins simultaneously output an active level for one or more cycles of the peripheral clock (Pϕ). (In the case of TOCS = 0 in timer output control register 1 (TOCR1) in the MTU2, low level when the output level select P (OLSP) bit is 0, or high level when the OLSP bit is 1. In the case of TOCS = 1, low level when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0 in TOCR2, or high level when these bits are 1.) This active level comparison is done when the MTU2 output function or general output function is selected in the pin function controller. If another function is selected, the output level is not checked. Pin combinations for output comparison and high-impedance control can be selected by POE2 registers.
PE5/PD10/PD29/TIOC3BS and PE6/PD11/PD28/TIOC3DS PE0/PD12/PD27/TIOC4AS and PE2/PD14/PD25/TIOC4CS PE1/PD13/PD26/TIOC4BS and PE3/PD15/PD24/TIOC4DS	Output	The high-current pins for the MTU2S are placed in high-impedance state when the pins simultaneously output an active level for one or more cycles of the peripheral clock (Pϕ). (In the case of TOCS = 0 in timer output control register 1S (TOCR1S) in the MTU2S, low level when the output level select P (OLSP) bit is 0, or high level when the OLSP bit is 1. In the case of TOCS = 1, low level when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits are 0 in TOCR2S, or high level when these bits are 1.) This active level comparison is done when the MTU2S output function or general output function is selected in the pin function controller. If another function is selected, the output level is not checked. Pin combinations for output comparison and high-impedance control can be selected by POE2 registers.

13.3 Register Descriptions

The POE2 has the following registers.

All these registers are initialized by a power-on reset, but are not initialized by a manual reset or in sleep mode, software standby mode, or module standby mode.

Table 13.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Input level control/status register 1	ICSR1	R/W	H'0000	H'FFFE5000	16
Output level control/status register 1	OCSR1	R/W	H'0000	H'FFFE5002	16
Input level control/status register 2	ICSR2	R/W	H'0000	H'FFFE5004	16
Output level control/status register 2	OCSR2	R/W	H'0000	H'FFFE5006	16
Input level control/status register 3	ICSR3	R/W	H'0000	H'FFFE5008	16
Software port output enable register	SPOER	R/W	H'00	H'FFFE500A	8
Port output enable control register 1	POECR1	R/W	H'00	H'FFFE500B	8
Port output enable control register 2	POECR2	R/W	H'7700	H'FFFE500C	16

13.3.1 Input Level Control/Status Register 1 (ICSR1)

ICSR1 is a 16-bit readable/writable register that selects the $\overline{POE0}$, $\overline{POE1}$, $\overline{POE2}$, and $\overline{POE3}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE3F	POE2F	POE1F	POE0F	-	-	-	PIE1	POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	POE3F	0	R/(W)*1	POE3 Flag Indicates that a high impedance request has been input to the $\overline{POE3}$ pin. [Clearing conditions] • By writing 0 to POE3F after reading POE3F = 1 (when the falling edge is selected by bits 7 and 6 in ICSR1) • By writing 0 to POE3F after reading POE3F = 1 after a high level input to $\overline{POE3}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 7 and 6 in ICSR1) [Setting condition] • When the input set by bits 7 and 6 in ICSR1 occurs at the $\overline{POE3}$ pin

Bit	Bit Name	Initial Value	R/W	Description
14	POE2F	0	R/(W)*[1]	POE2 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE2}}$ pin. [Clearing conditions] • By writing 0 to POE2F after reading POE2F = 1 (when the falling edge is selected by bits 5 and 4 in ICSR1) • By writing 0 to POE2F after reading POE2F = 1 after a high level input to $\overline{\text{POE2}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 5 and 4 in ICSR1) [Setting condition] • When the input set by bits 5 and 4 in ICSR1 occurs at the $\overline{\text{POE2}}$ pin
13	POE1F	0	R/(W)*[1]	POE1 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin. [Clearing conditions] • By writing 0 to POE1F after reading POE1F = 1 (when the falling edge is selected by bits 3 and 2 in ICSR1) • By writing 0 to POE1F after reading POE1F = 1 after a high level input to $\overline{\text{POE1}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 3 and 2 in ICSR1) [Setting condition] • When the input set by bits 3 and 2 in ICSR1 occurs at the $\overline{\text{POE1}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
12	POE0F	0	R/(W)*[1]	POE0 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin. [Clear conditions] • By writing 0 to POE0F after reading POE0F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR1) • By writing 0 to POE0F after reading POE0F = 1 after a high level input to $\overline{\text{POE0}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR1) [Set condition] • When the input set by bits 1 and 0 in ICSR1 occurs at the $\overline{\text{POE0}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PIE1	0	R/W	Port Interrupt Enable 1 Enables or disables interrupt requests when any one of the POE0F to POE3F bits of the ICSR1 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE3M[1:0]	00	R/W*[2]	POE3 Mode These bits select the input mode of the $\overline{\text{POE3}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE3}}$ input 01: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	POE2M[1:0]	00	R/W*[2]	POE2 Mode These bits select the input mode of the $\overline{\text{POE2}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE2}}$ input 01: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE2}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.
3, 2	POE1M[1:0]	00	R/W*[2]	POE1 Mode These bits select the input mode of the $\overline{\text{POE1}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE1}}$ input 01: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.
1, 0	POE0M[1:0]	00	R/W*[2]	POE0 Mode These bits select the input mode of the $\overline{\text{POE0}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE0}}$ input 01: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

13.3.2 Output Level Control/Status Register 1 (OCSR1)

OCSR1 is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF1	-	-	-	-	-	OCE1	OIE1	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	OSF1	0	R/(W)*1	Output Short Flag 1 Indicates that any one of the three pairs of MTU2 2-phase outputs 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 2-phase outputs has simultaneously become an active level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	OCE1	0	R/W*2	Output Short High-Impedance Enable 1 Specifies whether to place the pins in high-impedance state when the OSF1 bit in OCSR1 is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
8	OIE1	0	R/W	Output Short Interrupt Enable 1 Enables or disables interrupt requests when the OSF1 bit in OCSR is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

13.3.3 Input Level Control/Status Register 2 (ICSR2)

ICSR2 is a 16-bit readable/writable register that selects the $\overline{\text{POE4}}$ to $\overline{\text{POE7}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	POE4F	-	-	-	PIE2	-	-	-	-	-	-	POE4M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*1	R	R	R	R/W	R	R	R	R	R	R	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	POE4F	0	R/(W)*1	POE4 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE4}}$ pin. [Clearing conditions] • By writing 0 to POE4F after reading POE4F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR2) • By writing 0 to POE4F after reading POE4F = 1 after a high level input to $\overline{\text{POE4}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR2) [Setting condition] • When the input condition set by bits 1 and 0 in ICSR2 occurs at the $\overline{\text{POE4}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PIE2	0	R/W	Port Interrupt Enable 2 Enables or disables interrupt requests when the POE4F bit in the ICSR2 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	POE4M [1:0]	00	R/W*2	POE4 Mode These bits select the input mode of the $\overline{\text{POE4}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE4}}$ input 01: Accept request when $\overline{\text{POE4}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level. 10: Accept request when $\overline{\text{POE4}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level. 11: Accept request when $\overline{\text{POE4}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

13.3.4 Output Level Control/Status Register 2 (OCSR2)

OCSR2 is a 16-bit readable/writable register that controls the enable/disable of both output level comparison and interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF2	-	-	-	-	-	OCE2	OIE2	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	OSF2	0	R/(W)*1	Output Short Flag 2 Indicates that any one of the three pairs of MTU2S 2-phase outputs 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 2-phase outputs has simultaneously become an active level
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	OCE2	0	R/W*2	Output Short High-Impedance Enable 2 Specifies whether to place the pins in high-impedance state when the OSF2 bit in OCSR2 is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
8	OIE2	0	R/W	Output Short Interrupt Enable 2 Enables or disables interrupt requests when the OSF2 bit in OCSR2 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

13.3.5 Input Level Control/Status Register 3 (ICSR3)

ICSR3 is a 16-bit readable/writable register that selects the $\overline{\text{POE8}}$ pin input mode, controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	POE8F	-	-	POE8E	PIE3	-	-	-	-	-	-	POE8M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*1	R	R	R/W*2	R/W	R	R	R	R	R	R	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	POE8F	0	R/(W)*1	POE8 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE8}}$ pin. [Clearing conditions] • By writing 0 to POE8F after reading POE8F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR3) • By writing 0 to POE8F after reading POE8F = 1 after a high level input to $\overline{\text{POE8}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR3) [Setting condition] • When the input condition set by bits 1 and 0 in ICSR3 occurs at the $\overline{\text{POE8}}$ pin
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	POE8E	0	R/W*2	POE8 High-Impedance Enable Specifies whether to place the pins in high-impedance state when the POE8F bit in ICSR3 is set to 1. 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state
8	PIE3	0	R/W	Port Interrupt Enable 3 Enables or disables interrupt requests when the POE8F bit in ICSR3 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	POE8M[1:0]	00	R/W*2	POE8 Mode These bits select the input mode of the $\overline{\text{POE8}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE8}}$ input 01: Accept request when $\overline{\text{POE8}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE8}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE8}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

13.3.6 Software Port Output Enable Register (SPOER)

SPOER is an 8-bit readable/writable register that controls high-impedance state of the pins.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	MTU2SHIZ	0	R/W	MTU2S Output High-Impedance Specifies whether to place the high-current pins for the MTU2S in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2SHIZ after reading MTU2SHIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2SHIZ
1	MTU2CH0HIZ	0	R/W	MTU2 Channel 0 Output High-Impedance Specifies whether to place the pins for channel 0 in the MTU2 in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH0HIZ after reading MTU2CH0HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH0HIZ

Bit	Bit Name	Initial Value	R/W	Description
0	MTU2CH34HIZ	0	R/W	MTU2 Channels 3 and 4 Output High-Impedance Specifies whether to place the high-current pins for the MTU2 in high-impedance state. 0: Does not place the pins in high-impedance state [Clearing conditions] • Power-on reset • By writing 0 to MTU2CH34HIZ after reading MTU2CH34HIZ = 1 1: Places the pins in high-impedance state [Setting condition] • By writing 1 to MTU2CH34HIZ

13.3.7 Port Output Enable Control Register 1 (POECR1)

POECR1 is an 8-bit readable/writable register that controls high-impedance state of the pins.

Bit:	7	6	5	4	3	2	1	0
	MTU2 PB4ZE	MTU2 PB3ZE	MTU2 PB2ZE	MTU2 PB1ZE	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	MTU2PB4ZE	0	R/W*	MTU2PB4 High-Impedance Enable Specifies whether to place the PB4/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
6	MTU2PB3ZE	0	R/W*	MTU2PB3 High-Impedance Enable Specifies whether to place the PB3/TIOC0C pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
5	MTU2PB2ZE	0	R/W*	MTU2PB2 High-Impedance Enable Specifies whether to place the PB2/TIOC0B pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
4	MTU2PB1ZE	0	R/W*	MTU2PB1 High-Impedance Enable Specifies whether to place the PB1/TIOC0A pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
3	MTU2PE3ZE	0	R/W*	MTU2PE3 High-Impedance Enable Specifies whether to place the PE3/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
2	MTU2PE2ZE	0	R/W*	MTU2PE2 High-Impedance Enable Specifies whether to place the PE2/TIOC0C pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
1	MTU2PE1ZE	0	R/W*	MTU2PE1 High-Impedance Enable Specifies whether to place the PE1/TIOC0B pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
0	MTU2PE0ZE	0	R/W*	MTU2PE0 High-Impedance Enable Specifies whether to place the PE0/TIOC0A pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

Note: * Can modified only once after a power-on reset.

13.3.8 Port Output Enable Control Register 2 (POECR2)

POECR2 is a 16-bit readable/writable register that controls high-impedance state of the pins.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	-	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE	-	MTU2S P4CZE	MTU2S P5CZE	MTU2S P6CZE	-	MTU2S P7CZE	MTU2S P8CZE	MTU2S P9CZE
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*

Note: * Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	MTU2P1CZE	1	R/W*	MTU2 Port 1 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2 high-current PE9/TIOC3B and PE11/TIOC3D pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F to POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state 1: Compares output levels and places the pins in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
13	MTU2P2CZE	1	R/W*	MTU2 Port 2 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2 high-current PE12/TIOC4A and PE14/TIOC4C pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F to POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state 1: Compares output levels and places the pins in high-impedance state
12	MTU2P3CZE	1	R/W*	MTU2 Port 3 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2 high-current PE13/TIOC4B and PE15/TIOC4D pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F to POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state 1: Compares output levels and places the pins in high-impedance state
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	MTU2SP1CZE	1	R/W*	MTU2S Port 1 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PE5/TIOC3BS and PE6/TIOC3DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.

Bit	Bit Name	Initial Value	R/W	Description
9	MTU2SP2CZE	1	R/W*	MTU2S Port 2 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PE0/TIOC4AS and PE2/TIOC4CS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
8	MTU2SP3CZE	1	R/W*	MTU2S Port 3 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PE1/TIOC4BS and PE3/TIOC4DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	MTU2SP4CZE	0	R/W*	MTU2S Port 4 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PD10/TIOC3BS and PD11/TIOC3DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
5	MTU2SP5CZE	0	R/W*	MTU2S Port 5 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PD12/TIOC4AS and PD14/TIOC4CS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
4	MTU2SP6CZE	0	R/W*	MTU2S Port 6 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PD13/TIOC4BS and PD15/TIOC4DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MTU2SP7CZE	0	R/W*	MTU2S Port 7 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PD29/TIOC3BS and PD28/TIOC3DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
1	MTU2SP8CZE	0	R/W*	MTU2S Port 8 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PD27/TIOC4AS and PD25/TIOC4CS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.

Bit	Bit Name	Initial Value	R/W	Description
0	MTU2SP9CZE	0	R/W*	MTU2S Port 9 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PD26/TIOC4BS and PD24/TIOC4DS pins and to place them in high-impedance state when the OSF2 bit is set to 1 while the OCE2 bit is 1 or when one of the POE4F and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.

Note: * Can be modified only once after a power-on reset.

13.4 Operation

Table 13.4 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 13.4 Target Pins and Conditions for High-Impedance Control

Pins	Conditions	Detailed Conditions
MTU2 high-current pins (PE9/TIOC3B and PE11/TIOC3D)	Input level detection, output level comparison, or SPOER setting	MTU2P1CZE ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PE12/TIOC4A and PE14/TIOC4C)	Input level detection, output level comparison, or SPOER setting	MTU2P2CZE ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PE13/TIOC4B and PE15/TIOC4D)	Input level detection, output level comparison, or SPOER setting	MTU2P3CZE ((POE3F+POE2F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2S high-current pins (PE5/TIOC3BS and PE6/TIOC3DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP1CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PE0/TIOC4A and PE2/TIOC4CS)	Input level detection, output level comparison, or SPOER setting	MTU2SP2CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PE1/TIOC4BS and PE3/TIOC4DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP3CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD10/TIOC3BS and PD11/TIOC3DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP4CZE (POE4F +(OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD12/TIOC4AS and PD14/TIOC4CS)	Input level detection, output level comparison, or SPOER setting	MTU2SP5CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD13/TIOC4BS and PD15/TIOC4DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP6CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD29/TIOC3BS and PD28/TIOC3DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP7CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))

Pins	Conditions	Detailed Conditions
MTU2S high-current pins (PD27/TIOC4AS and PD25/TIOC4CS)	Input level detection, output level comparison, or SPOER setting	MTU2SP8CZE (POE4F +(OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD26/TIOC4BS and PD24/TIOC4DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP9CZE (POE4F + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2 CH0 pins (PE0/TIOC0A, PE1/TIOC0B, PE2/TIOC0C, and PE3/TIOC0D)	Input level detection or SPOER setting	MTU2PE0ZE to MTU2PE3ZE (POE8F • POE8E) +(MTU2CH0HIZ)
MTU2 CH0 pins (PB1/TIOC0A, PB2/TIOC0B, PB3/TIOC0C, and PB4/TIOC0D)	Input level detection or SPOER setting	MTU2PB1ZE to MTU2PB4ZE (POE8F • POE8E) +(MTU2CH0HIZ)

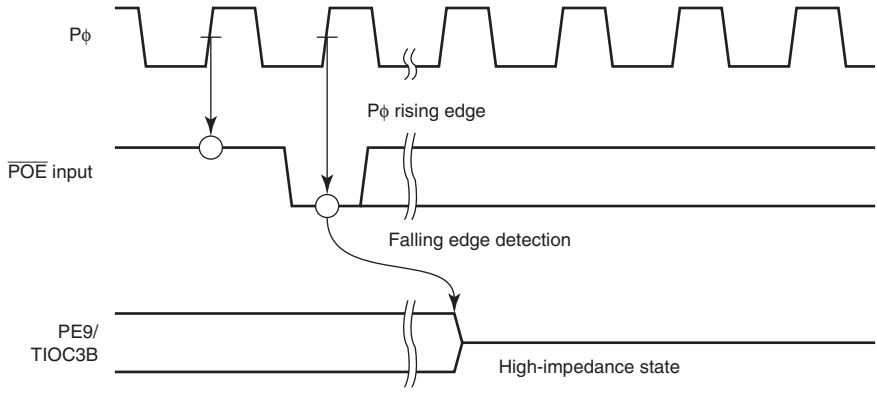
13.4.1 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR3 occur on the $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins, the high-current pins and the pins for channel 0 of the MTU2 are placed in high-impedance state. Note however, that these high-current and MTU2 pins enter high-impedance state only when general input/output function, MTU2 function, or MTU2S function is selected for these pins.

(1) Falling Edge Detection

When a change from a high to low level is input to the $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins, the high-current pins and the pins for channel 0 of the MTU2 are placed in high-impedance state.

Figure 13.2 shows the sample timing after the level changes in input to the $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins until the respective pins enter high-impedance state.



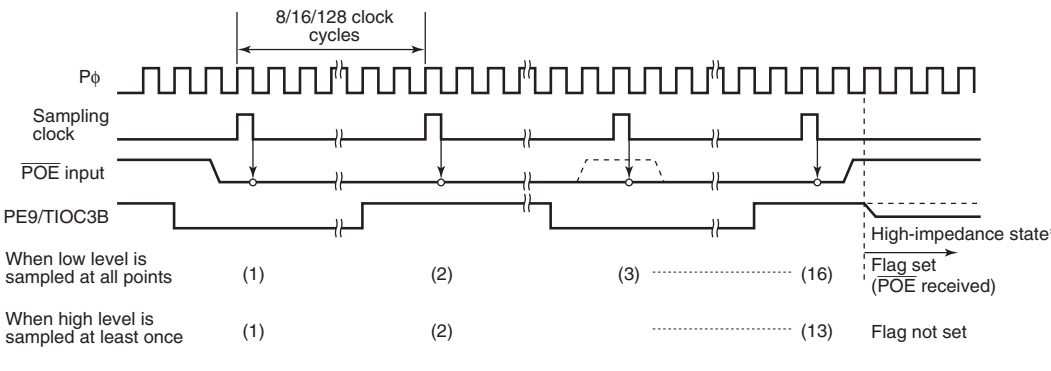
Note: The other high-current pins and MTU2 channel 0 pins also enter the high-impedance state in the similar timing.

Figure 13.2 Falling Edge Detection

(2) Low-Level Detection

Figure 13.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 to ICSR3. If even one high level is detected during this interval, the low level is not accepted.

The timing when the high-current pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.



Note: * The other high-current pins and MTU2 channel 0 pins also enter the high-impedance state in the similar timing.

Figure 13.3 Low-Level Detection Operation

13.4.2 Output-Level Compare Operation

Figure 13.4 shows an example of the output-level compare operation for the combination of TIOC3B and TIOC3D. The operation is the same for the other pin combinations.

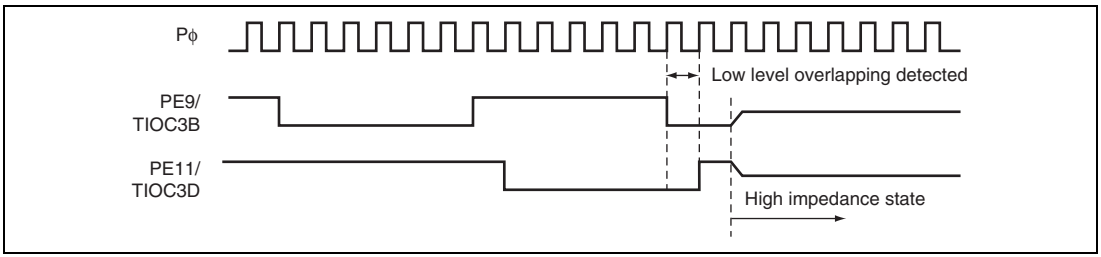


Figure 13.4 Output-Level Compare Operation

13.4.3 Release from High-Impedance State

High-current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the flags in bits 15 to 12 (POE8F, POE4F to POE0F) of ICSR1 to ICSR3. However, note that when low-level sampling is selected by bits 7 to 0 in ICSR1 to ICSR3, 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 one of the $\overline{\text{POE0}}$ to $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins and is sampled.

High-current pins that have entered high-impedance state due to output-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing the flag in bit 15 (OCF1 and OCF2) 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 high-current pins. Inactive-level outputs can be achieved by setting the MTU2 and MTU2S internal registers.

13.5 Interrupts

The POE2 issues a request to generate an interrupt when the specified condition is satisfied during input level detection or output level comparison. Table 13.5 shows the interrupt sources and their conditions.

Table 13.5 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE3F, POE2F, POE1F, POE0F, and OSF1	PIE1 • (POE3F + POE2F + POE1F + POE0F) + OIE1 • OSF1
OEI2	Output enable interrupt 2	POE8F	PIE3 • POE8F
OEI3	Output enable interrupt 3	POE4F and OSF2	PIE2 • POE4F + OIE2 • OSF2

13.6 Usage Notes

13.6.1 Pins States when the Watchdog Timer has Issued a Power-on Reset

A power-on reset issued from the watchdog timer (WDT) initializes the pin-function controller (PFC) and all I/O port pins thus become general-purpose inputs in accord with the initial PFC settings. However, when a power-on reset is issued while the port-output enable (POE) setting is for high-impedance handling by the pins, the pins remain in the output state for an interval of one cycle of the peripheral clock (Pϕ) before switching to operation as general-purpose inputs.

The same condition applies when the WDT issues a power-on reset and short-circuit detection by the MTU2 has led to high-impedance handling by a pin.

Figure 13.5 shows the situation where timer output has been selected and the WDT issues a power-on reset while high-impedance handling is in progress due to the $\overline{\text{POE}}$ input.

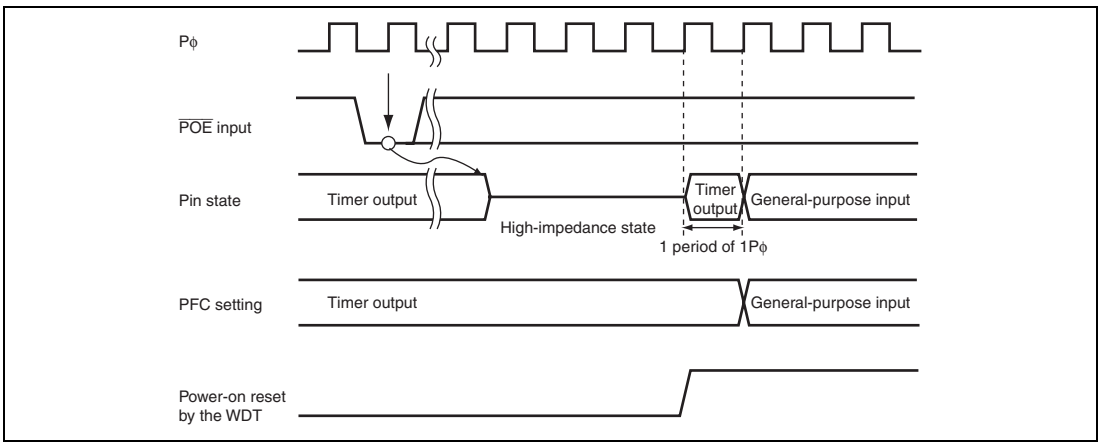


Figure 13.5 Pin States when the Watchdog Timer Issues a Power-on Reset

13.6.2 Input Pins

When the POE function is to be used, input a logical 1 to the POE0 to POE4 and POE8 pins by the time the PFC is set for POE input.

Section 14 Compare Match Timer (CMT)

This LSI has an on-chip compare match timer (CMT) consisting of a two-channel 16-bit timer. The CMT has a16-bit counter, and can generate interrupts at set intervals.

14.1 Features

- Independent selection of four counter input clocks at two channels
  Any of four internal clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) can be selected.
- Selection of DTC/DMA transfer request or interrupt request generation on compare match by DTC/DMA setting
- When not in use, the CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 14.1 shows a block diagram of CMT.

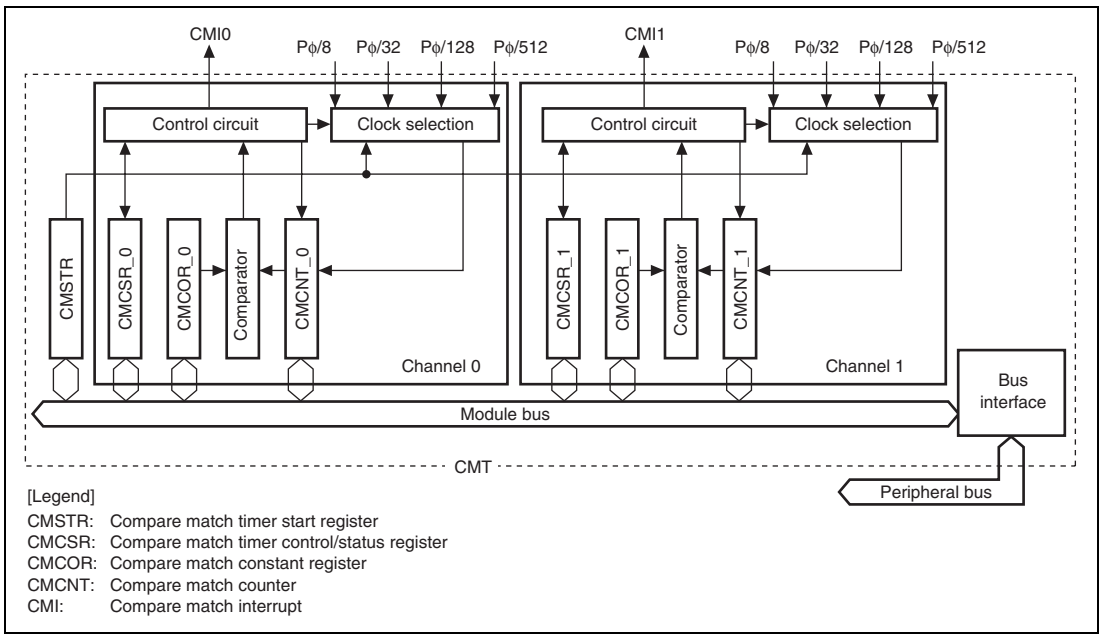


Figure 14.1 Block Diagram of CMT

14.2 Register Descriptions

The CMT has the following registers.

Table 14.1 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFEC000	16
0	Compare match timer control/ status register_0	CMCSR_0	R/(W)*	H'0000	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	R/W	H'0000	H'FFFEC004	16
	Compare match constant register_0	CMCOR_0	R/W	H'FFFF	H'FFFEC006	16
1	Compare match timer control/ status register_1	CMCSR_1	R/(W)*	H'0000	H'FFFEC008	16
	Compare match counter_1	CMCNT_1	R/W	H'0000	H'FFFEC00A	16
	Compare match constant register_1	CMCOR_1	R/W	H'FFFF	H'FFFEC00C	16

14.2.1 Compare Match Timer Start Register (CMSTR)

CMSTR is a 16-bit register that selects whether compare match counter (CMCNT) operates or is stopped.

CMSTR is initialized to H'0000 by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	STR1	0	R/W	Count Start 1 Specifies whether compare match counter_1 operates or is stopped. 0: CMCNT_1 count is stopped 1: CMCNT_1 count is started
0	STR0	0	R/W	Count Start 0 Specifies whether compare match counter_0 operates or is stopped. 0: CMCNT_0 count is stopped 1: CMCNT_0 count is started

14.2.2 Compare Match Timer Control/Status Register (CMCSR)

CMCSR is a 16-bit register that indicates compare match generation, enables or disables interrupts, and selects the counter input clock.

CMCSR is initialized to H'0000 by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match. [Clearing condition] • When 0 is written to CMF after reading CMF = 1 • When data is transferred after the DTC has been activated by CMI (except when the DTC transfer counter value has become H'000). • When data is transferred after the DMAC has been activated by CMI 1: CMCNT and CMCOR values match
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled

Bit	Bit Name	Initial Value	R/W	Description
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select These bits select the clock to be input to CMCNT from four internal clocks obtained by dividing the peripheral clock (Pϕ). When the STR bit in CMSTR is set to 1, CMCNT starts counting on the clock selected with bits CKS[1:0]. 00: Pϕ/8 01: Pϕ/32 10: Pϕ/128 11: Pϕ/512

Note: * Only 0 can be written to clear the flag after 1 is read.

14.2.3 Compare Match Counter (CMCNT)

CMCNT is a 16-bit register used as an up-counter. When the counter input clock is selected with bits CKS[1:0] in CMCSR, and the STR bit in CMSTR is set to 1, CMCNT starts counting using the selected clock. When the value in CMCNT and the value in compare match constant register (CMCOR) match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1.

CMCNT is initialized to H'0000 by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.2.4 Compare Match Constant Register (CMCOR)

CMCOR is a 16-bit register that sets the interval up to a compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset or in module standby mode, but retains its previous value in software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14.3 Operation

14.3.1 Interval Count Operation

When an internal clock is selected with the CKS[1:0] bits in CMCSR and the STR bit in CMSTR is set to 1, CMCNT starts incrementing using the selected clock. When the values in CMCNT and CMCOR match, CMCNT is cleared to H'0000 and the CMF flag in CMCSR is set to 1. When the CMIE bit in CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested. CMCNT then starts counting up again from H'0000.

Figure 14.2 shows the operation of the compare match counter.

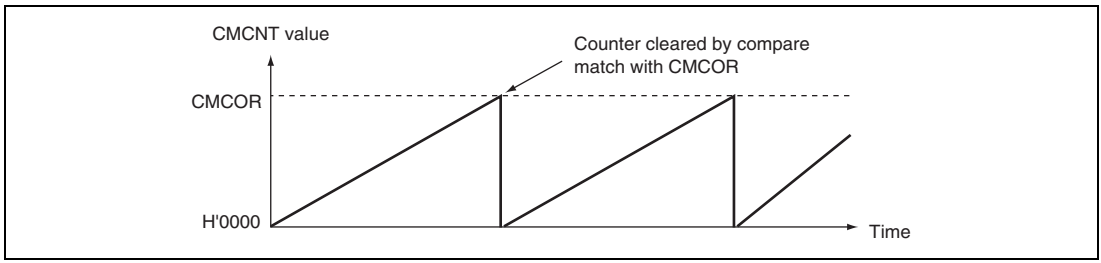


Figure 14.2 Counter Operation

14.3.2 CMCNT Count Timing

One of four clocks (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) obtained by dividing the peripheral clock (Pϕ) can be selected with the CKS[1:0] bits in CMCSR. Figure 14.3 shows the timing.

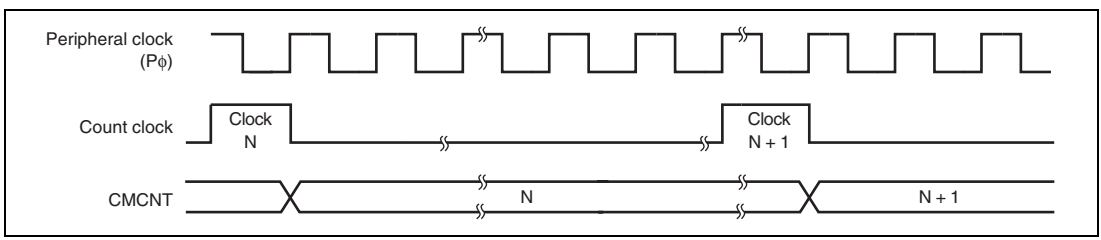


Figure 14.3 Count Timing

14.4 Interrupts

14.4.1 Interrupt Sources and DTC/DMAC Transfer Requests

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt. When both the interrupt request flag (CMF) and the interrupt enable bit (CMIE) are set to 1, the corresponding interrupt request is output. When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 6, Interrupt Controller (INTC).

Clear the CMF bit to 0 by the user exception handling routine. If this operation is not carried out, another interrupt will be generated. The direct memory access controller (DMAC) can be set to be activated when a compare match interrupt is requested. In this case, an interrupt is not issued to the CPU. If the setting to activate the DMAC has not been made, an interrupt request is sent to the CPU. The CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

The data transfer controller (DTC) can be activated by an interrupt request. In this case, the priority between channels is fixed. For details, refer to section 8, Data Transfer Controller (DTC).

Table 14.2 Interrupt Sources

Channel	Interrupt Source	Interrupt Enable Bit	Interrupt Flag	DMAC/DTC Activation	Priority
0	CMI0	CMIE	CMF	Possible	High
1	CMI1	CMIE	CMF	Possible	Low

14.4.2 Timing of Compare Match Flag Setting

When CMCOR and CMCNT match, a compare match signal is generated at the last state in which the values match (the timing when the CMCNT value is updated to H'0000) and the CMF bit in CMCSR is set to 1. That is, after a match between CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT counter clock input. Figure 14.4 shows the timing of CMF bit setting.

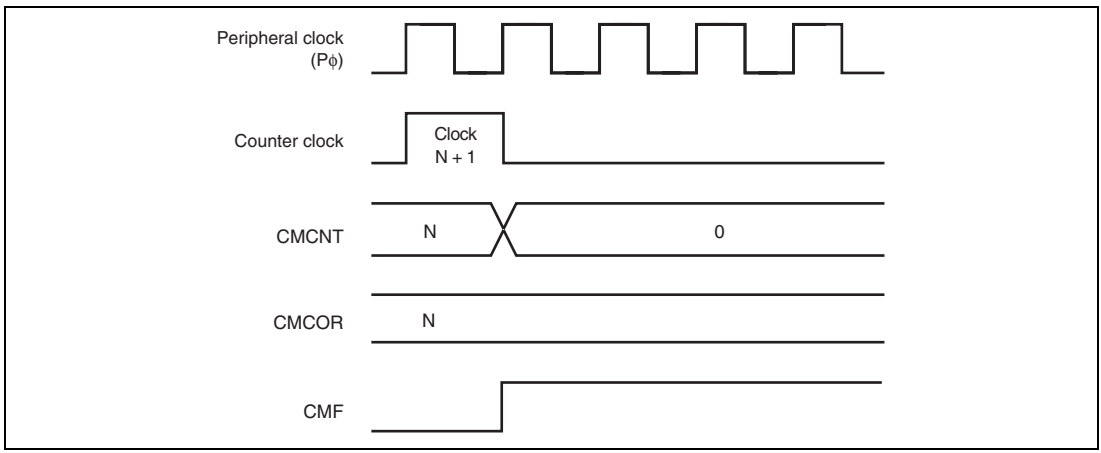


Figure 14.4 Timing of CMF Setting

14.4.3 Timing of Compare Match Flag Clearing

The CMF bit in CMCSR is cleared by first, reading as 1 then writing to 0. However, in the case of the DMAC being activated, the CMF bit is automatically cleared to 0 when data is transferred by the DMAC.

14.5 Usage Notes

14.5.1 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated in the T2 cycle while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 14.5 shows the timing to clear the CMCNT counter.

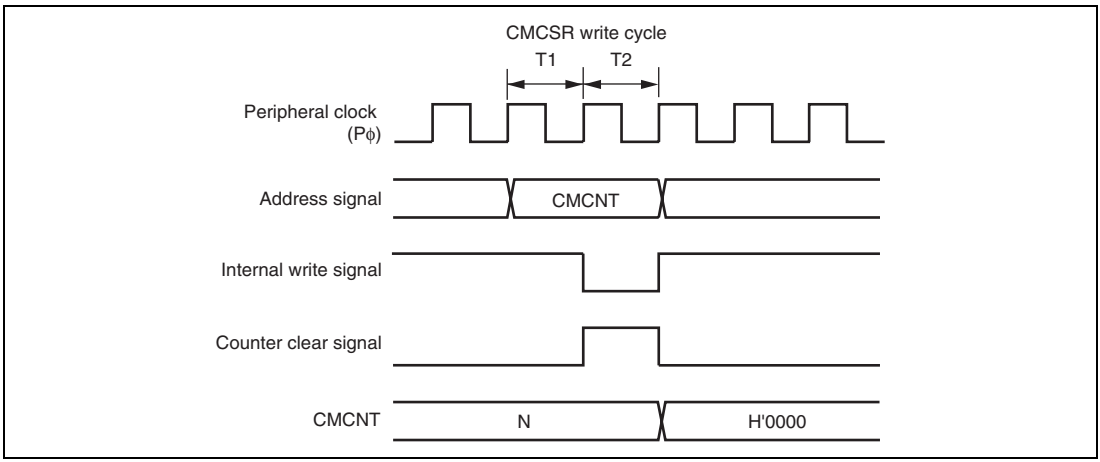


Figure 14.5 Conflict between Write and Compare Match Processes of CMCNT

14.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in words, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 14.6 shows the timing to write to CMCNT in words.

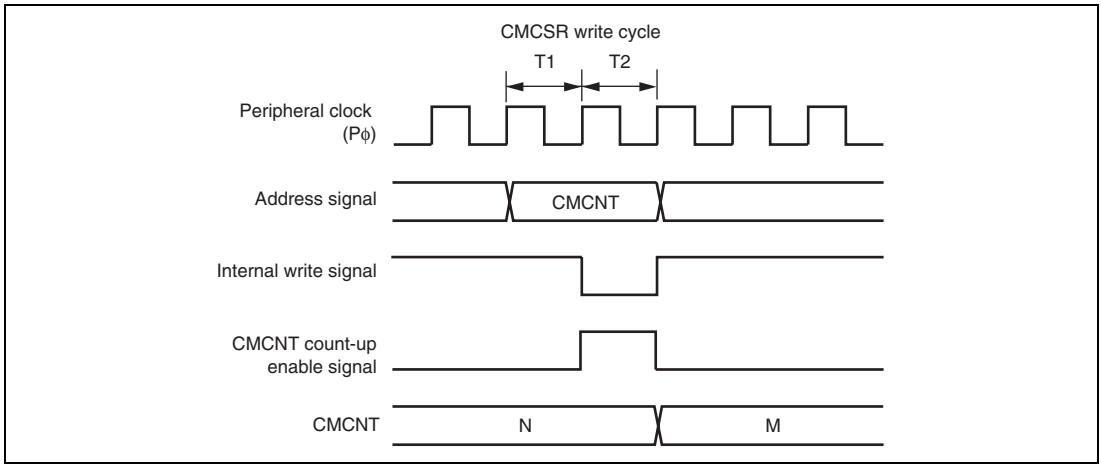


Figure 14.6 Conflict between Word-Write and Count-Up Processes of CMCNT

14.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT

Even when the count-up occurs in the T2 cycle while writing to CMCNT in bytes, the writing has priority over the count-up. In this case, the count-up is not performed. The byte data on the other side, which is not written to, is also not counted and the previous contents are retained.

Figure 14.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNTH in bytes.

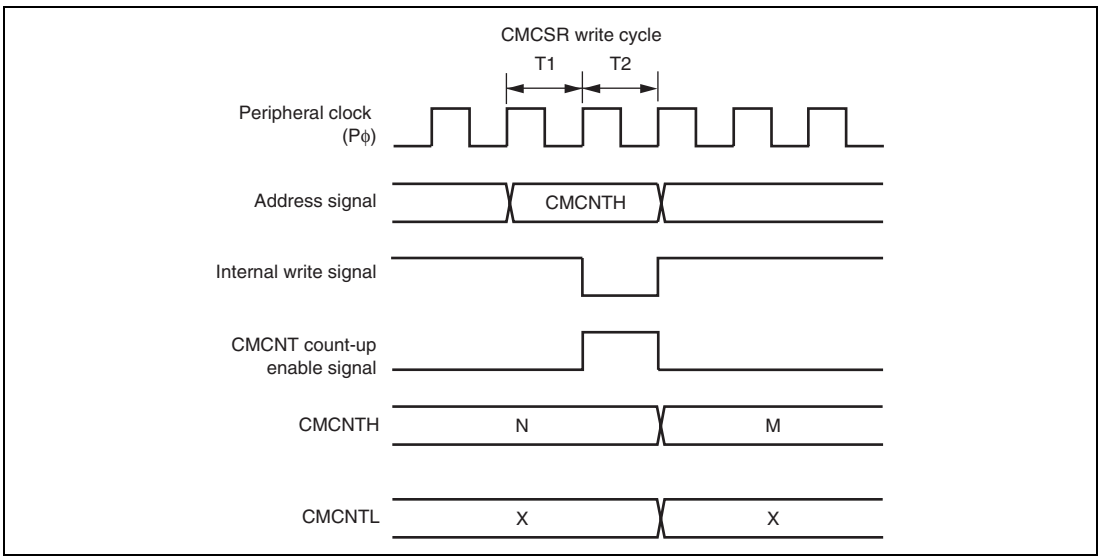


Figure 14.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

14.5.4 Compare Match between CMCNT and CMCOR

Do not set a same value to CMCNT and CMCOR while the count operation of CMCNT is stopped.

Section 15 Watchdog Timer (WDT)

This LSI includes the watchdog timer (WDT), which externally outputs an overflow signal ($\overline{\text{WDTOVF}}$) on overflow of the counter when the value of the counter has not been updated because of a system malfunction. The WDT can simultaneously generate an internal reset signal for the entire LSI.

The WDT is a single channel timer that counts up the clock oscillation settling period when the system leaves the temporary standby periods that occur when the clock frequency is changed. It can also be used as a general watchdog timer or interval timer.

15.1 Features

- Can be used to ensure the clock oscillation settling time
  The WDT is used in leaving the temporary standby periods that occur when the clock frequency is changed.
- Can switch between watchdog timer mode and interval timer mode.
- Outputs $\overline{\text{WDTOVF}}$ signal in watchdog timer mode
  When the counter overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. It is possible to select whether to reset the LSI internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset type.
- Interrupt generation in interval timer mode
  An interval timer interrupt is generated when the counter overflows.
- Choice of eight counter input clocks
  Eight clocks ($P\phi \times 1$ to $P\phi \times 1/16384$) that are obtained by dividing the peripheral clock can be selected.

Figure 15.1 shows a block diagram of the WDT.

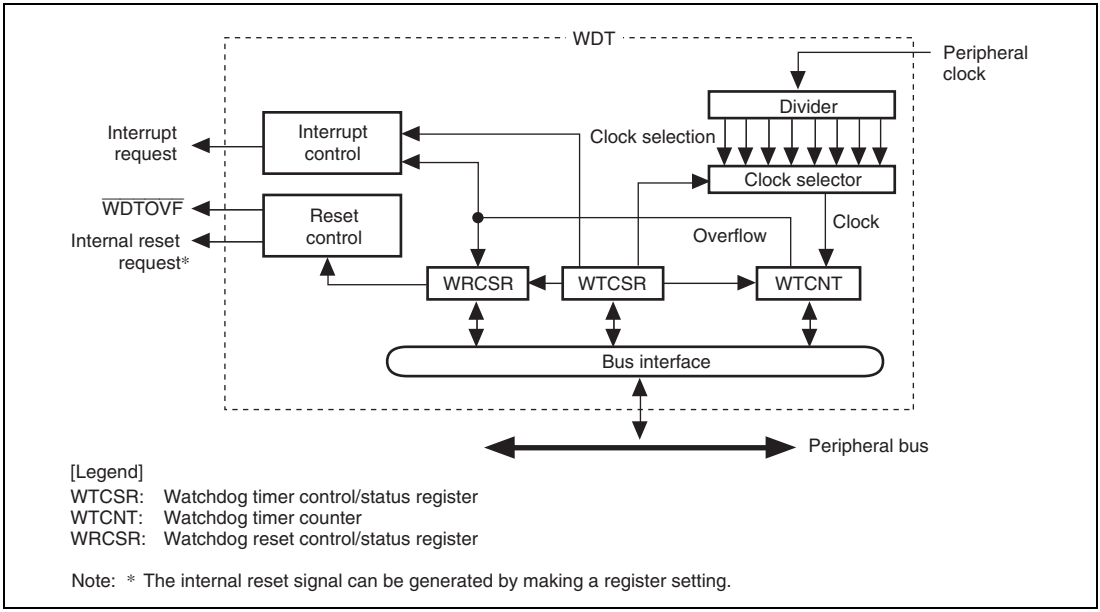


Figure 15.1 Block Diagram of WDT

15.2 Input/Output Pin

Table 15.1 shows the pin configuration of the WDT.

Table 15.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

15.3 Register Descriptions

The WDT has the following registers.

Table 15.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Watchdog timer counter	WTCNT	R/W	H'00	H'FFFE0002	16*
Watchdog timer control/status register	WTCSR	R/W	H'18	H'FFFE0000	16*
Watchdog reset control/status register	WRCSR	R/W	H'1F	H'FFFE0004	16*

Note: * For the access size, see section 15.3.4, Notes on Register Access.

15.3.1 Watchdog Timer Counter (WTCNT)

WTCNT is an 8-bit readable/writable register that is incremented by cycles of the selected clock signal. When an overflow occurs, it generates a watchdog timer overflow signal ($\overline{\text{WDTOVF}}$) in watchdog timer mode and an interrupt in interval timer mode.

WTCNT is initialized to H'00 by a power-on reset caused by the $\overline{\text{RES}}$ pin or in software standby mode.

Use word access to write to WTCNT, writing H'5A in the upper byte. Use byte access to read from WTCNT.

Note: The method for writing to WTCNT differs from that for other registers to prevent erroneous writes. See section 15.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15.3.2 Watchdog Timer Control/Status Register (WTCSR)

WTCSR is an 8-bit readable/writable register composed of bits to select the clock used for the count, overflow flags, and timer enable bit.

WTCSR is initialized to H'18 by a power-on reset caused by the $\overline{\text{RES}}$ pin, an internal reset caused by the WDT, or in software standby mode.

Use word access to write to WTCSR, writing H'A5 in the upper byte. Use byte access to read from WTCSR.

Note: The method for writing to WTCSR differs from that for other registers to prevent erroneous writes. See section 15.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{\text{IT}}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IOVF	0	R/(W)	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: No overflow 1: WTCNT overflow in interval timer mode [Clearing condition] • When 0 is written to IOVF after reading IOVF
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: When the WTCNT overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. If this bit is modified when the WDT is running, the up-count may not be performed correctly.

Bit	Bit Name	Initial Value	R/W	Description
5	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in software standby mode or when changing the clock frequency. 0: Timer disabled Count-up stops and WTCNT value is retained 1: Timer enabled
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2 to 0	CKS[2:0]	000	R/W	Clock Select These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow period that is shown in the table is the value when the peripheral clock (Pϕ) is 40 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 × Pϕ	6.4 µs
001:	1/64 × Pϕ	409.6 µs
010:	1/128 × Pϕ	819.2 ms
011:	1/256 × Pϕ	1.64 ms
100:	1/512 × Pϕ	3.3 ms
101:	1/1024 × Pϕ	6.6 ms
110:	1/4096 × Pϕ	26.2 ms
111:	1/16384 × Pϕ	104.9 ms

Note: If bits CKS[2:0] are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

15.3.3 Watchdog Reset Control/Status Register (WRCSR)

WRCSR is an 8-bit readable/writable register that controls output of the internal reset signal generated by watchdog timer counter (WTCNT) overflow.

WRCSR is initialized to H'1F by input of a reset signal from the $\overline{\text{RES}}$ pin, but is not initialized by the internal reset signal generated by overflow of the WDT. WRCSR is initialized to H'1F in software standby mode.

Note: The method for writing to WRCSR differs from that for other registers to prevent erroneous writes. See section 15.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)	R/W	R/W	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)	Watchdog Timer Overflow Indicates that the WTCNT has overflowed in watchdog timer mode. This bit is not set in interval timer mode. 0: No overflow 1: WTCNT has overflowed in watchdog timer mode [Clearing condition] • When 0 is written to WOVF after reading WOVF
6	RSTE	0	R/W	Reset Enable Selects whether to generate a signal to reset the LSI internally if WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Not reset when WTCNT overflows* 1: Reset when WTCNT overflows Note: * LSI not reset internally, but WTCNT and WTCSR reset within WDT.

Bit	Bit Name	Initial Value	R/W	Description
5	RSTS	0	R/W	Reset Select Selects the type of reset when the WTCNT overflows in watchdog timer mode. In interval timer mode, this setting is ignored. 0: Power-on reset 1: Manual reset
4 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

15.3.4 Notes on Register Access

The watchdog timer counter (WTCNT), watchdog timer control/status register (WTCSR), and watchdog reset control/status register (WRCSR) are more difficult to write to than other registers. The procedures for reading or writing to these registers are given below.

(1) Writing to WTCNT and WTCSR

These registers must be written by a word transfer instruction. They cannot be written by a byte or longword transfer instruction.

When writing to WTCNT, set the upper byte to H'5A and transfer the lower byte as the write data, as shown in figure 15.2. When writing to WTCSR, set the upper byte to H'A5 and transfer the lower byte as the write data. This transfer procedure writes the lower byte data to WTCNT or WTCSR.

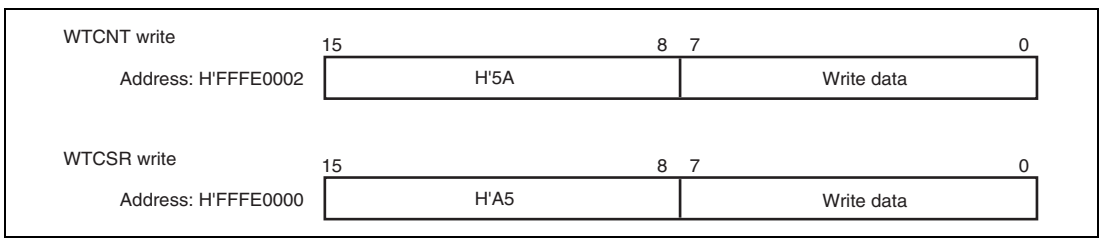


Figure 15.2 Writing to WTCNT and WTCSR

(2) Writing to WRCSR

WRCSR must be written by a word access to address H'FFFE0004. It cannot be written by byte transfer or longword transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 15.3.

To write 0 to the WOVF bit, write H'A5 to the upper byte and write the write data to the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.

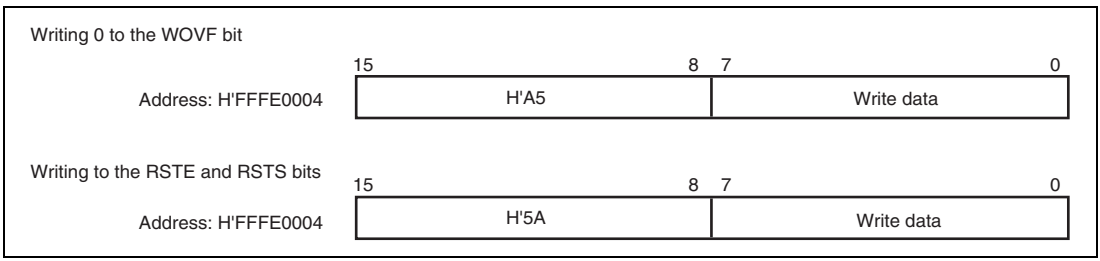


Figure 15.3 Writing to WRCSR

(3) Reading from WTCNT, WTCSR, and WRCSR

WTCNT, WTCSR, and WRCSR are read in a method similar to other registers. WTCSR is allocated to address H'FFFE0000, WTCNT to address H'FFFE0002, and WRCSR to address H'FFFE0004. Byte transfer instructions must be used for reading from these registers.

15.4 WDT Usage

15.4.1 Canceling Software Standby Mode

The WDT can be used to cancel software standby mode with an interrupt such as an NMI interrupt. The procedure is described below. (The WDT does not operate when resets are used for canceling, so keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until clock oscillation settles.)

1. Before making a transition to software standby mode, always clear the TME bit in WTCSR to 0. When the TME bit is 1, an erroneous reset or interval timer interrupt may be generated when the count overflows.
2. Set the type of count clock used in the CKS[2:0] bits in WTCSR and the initial value of the counter in WTCNT. These values should ensure that the time till count overflow is longer than the clock oscillation settling time.
3. After setting the STBY bit of the standby control register (STBCR: see section 30, Power-Down Modes) to 1, the execution of a SLEEP instruction puts the system in software standby mode and clock operation then stops.
4. The WDT starts counting by detecting the edge change of the NMI signal.
5. When the WDT count overflows, the CPG starts supplying the clock and this LSI resumes operation. The WOVF flag in WRCSR is not set when this happens.

15.4.2 Using Watchdog Timer Mode

1. Set the WT/$\overline{\text{IT}}$ bit in WTCSR to 1, the type of count clock in the CKS[2:0] bits in WTCSR, whether this LSI is to be reset internally or not in the RSTE bit in WRCSR, the reset type if it is generated in the RSTS bit in WRCSR, and the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in watchdog timer mode.
3. While operating in watchdog timer mode, rewrite the counter periodically to H'00 to prevent the counter from overflowing.
4. When the counter overflows, the WDT sets the WOVF flag in WRCSR to 1, and the $\overline{\text{WDTOVF}}$ signal is output externally (figure 15.4). The $\overline{\text{WDTOVF}}$ signal can be used to reset the system. The $\overline{\text{WDTOVF}}$ signal is output for $64 \times P\phi$ clock cycles.
5. If the RSTE bit in WRCSR is set to 1, a signal to reset the inside of this LSI can be generated simultaneously with the $\overline{\text{WDTOVF}}$ signal. Either power-on reset or manual reset can be selected for this interrupt by the RSTS bit in WRCSR. The internal reset signal is output for $128 \times P\phi$ clock cycles.

6. When a WDT overflow reset is generated simultaneously with a reset input on the $\overline{\text{RES}}$ pin, the $\overline{\text{RES}}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.
7. Since WTCSR is initialized by an internal reset caused by the WDT, the TME bit in WTCSR is cleared to 0. This makes the counter stop (be initialized). To use the WDT in watchdog timer mode again, after clearing the WOVF flag in WRCSR, set watchdog timer mode again.

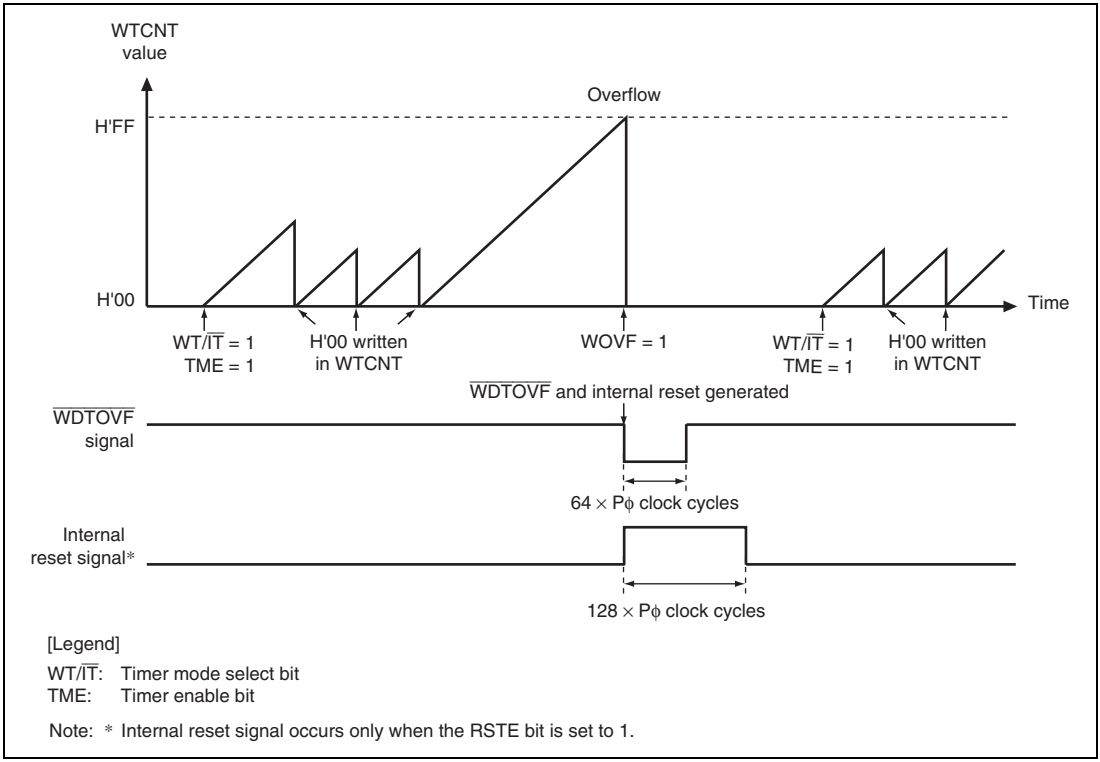


Figure 15.4 Operation in Watchdog Timer Mode

15.4.3 Using Interval Timer Mode

When operating in interval timer mode, interval timer interrupts are generated at every overflow of the counter. This enables interrupts to be generated at set periods.

1. Clear the WT/$\overline{IT}$ bit in WTCSR to 0, set the type of count clock in the CKS[2:0] bits in WTCSR, and set the initial value of the counter in WTCNT.
2. Set the TME bit in WTCSR to 1 to start the count in interval timer mode.
3. When the counter overflows, the WDT sets the IOVF bit in WTCSR to 1 and an interval timer interrupt request is sent to the INTC. The counter then resumes counting.

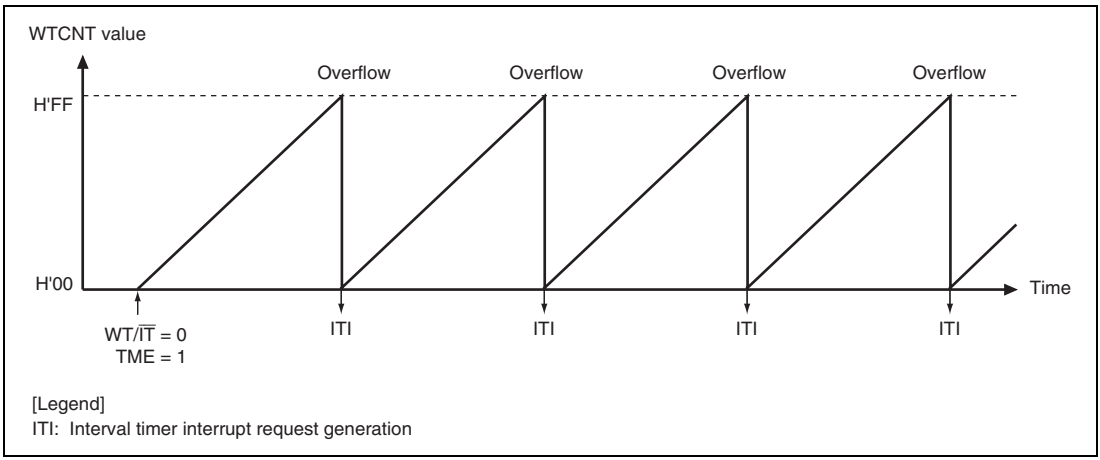


Figure 15.5 Operation in Interval Timer Mode

15.5 Interrupt Sources

The watchdog timer has the interval timer interrupt (ITI).

Table 15.3 gives details on the interrupt source.

The interval timer interrupt (ITI) is generated when the interval timer overflow flag (IOVF) in the watchdog timer control/status register (WTCSR) is set to 1.

Clearing the interrupt flag bit to 0 cancels the interrupt request.

Table 15.3 Interrupt Source

Abbreviation	Interrupt Source	Interrupt Enable Bit	Interrupt Flag
ITI	Interval timer interrupt	—	Interval timer overflow flag (IOVF)

15.6 Usage Notes

Pay attention to the following points when using the WDT in either the interval timer or watchdog timer mode.

15.6.1 Timer Variation

After timer operation has started, the period from the power-on reset point to the first count up timing of WTCNT varies depending on the time period that is set by the TME bit of WTCSR. The shortest such time period is thus one cycle of the peripheral clock, Pϕ, while the longest is the result of frequency division according to the value in the CKS[2:0] bits. The timing of subsequent incrementation is in accord with the selected frequency division ratio. Accordingly, this time difference is referred to as timer variation.

This also applies to the timing of the first incrementation after WTCNT has been written to during timer operation.

15.6.2 Prohibition against Setting H'FF to WTCNT

When the value in WTCNT reaches H'FF, the WDT assumes that an overflow has occurred. Accordingly, when H'FF is set in WTCNT, an interval timer interrupt or WDT reset will occur immediately, regardless of the current clock selection by the CKS[2:0] bits.

15.6.3 Interval Timer Overflow Flag

As long as the value of WTCNT is H'FF, clearing the IOVF flag in WTCSR is not possible. Clear the flag when the value of WTCNT becomes H'00 or after writing a value other than H'FF to WTCNT.

15.6.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circuits. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 15.6.

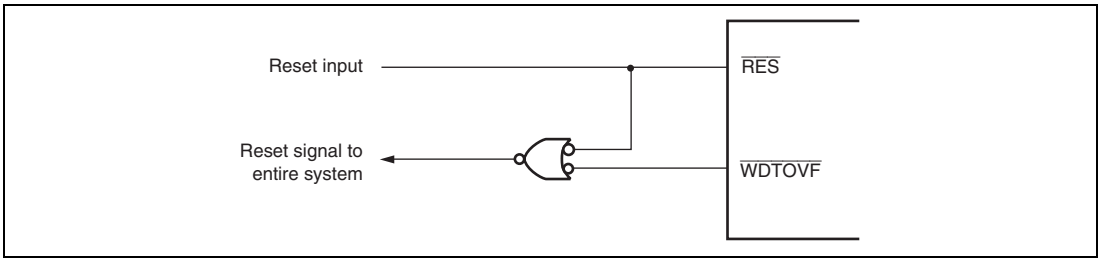


Figure 15.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

15.6.5 Manual Reset in Watchdog Timer Mode

When a manual reset occurs in watchdog timer mode, the intermal bus (I bus) cycle is continued. If a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception handling will be pended until the CPU acquires the bus mastership.

15.6.6 Connection of the $\overline{\text{WDTOVF}}$ Pin

When the $\overline{\text{WDTOVF}}$ pin is not in use, leave the pin open-circuit. If pulling down is required, the value of the resistor must be at least 1 MΩ.

Section 16 Serial Communication Interface (SCI)

This LSI has four channels of independent serial communication interface (SCI). The SCI can handle both asynchronous and clock synchronous serial communication. In asynchronous serial communication mode, serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function).

16.1 Features

- Choice of asynchronous or clock synchronous serial communication mode
- Asynchronous mode:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communications interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are twelve selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Multiprocessor communications
  - Receive error detection: Parity, overrun, and framing errors
  - Break detection: Break is detected by reading the RXD pin level directly when a framing error occurs.
- Clock synchronous mode:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clock synchronous communication function.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal clock) or SCK pin (external clock)
- Choice of LSB-first or MSB-first data transfer (except for 7-bit data in asynchronous mode)

- Four types of interrupts: There are four interrupt sources, transmit-data-empty, transmit end, receive-data-full, and receive error interrupts, and each interrupt can be requested independently. The data transfer controller (DTC) can be activated by the transmit-data-empty interrupt or receive-data-full interrupt to transfer data.
- Module standby mode can be set

Figure 16.1 shows a block diagram of the SCI.

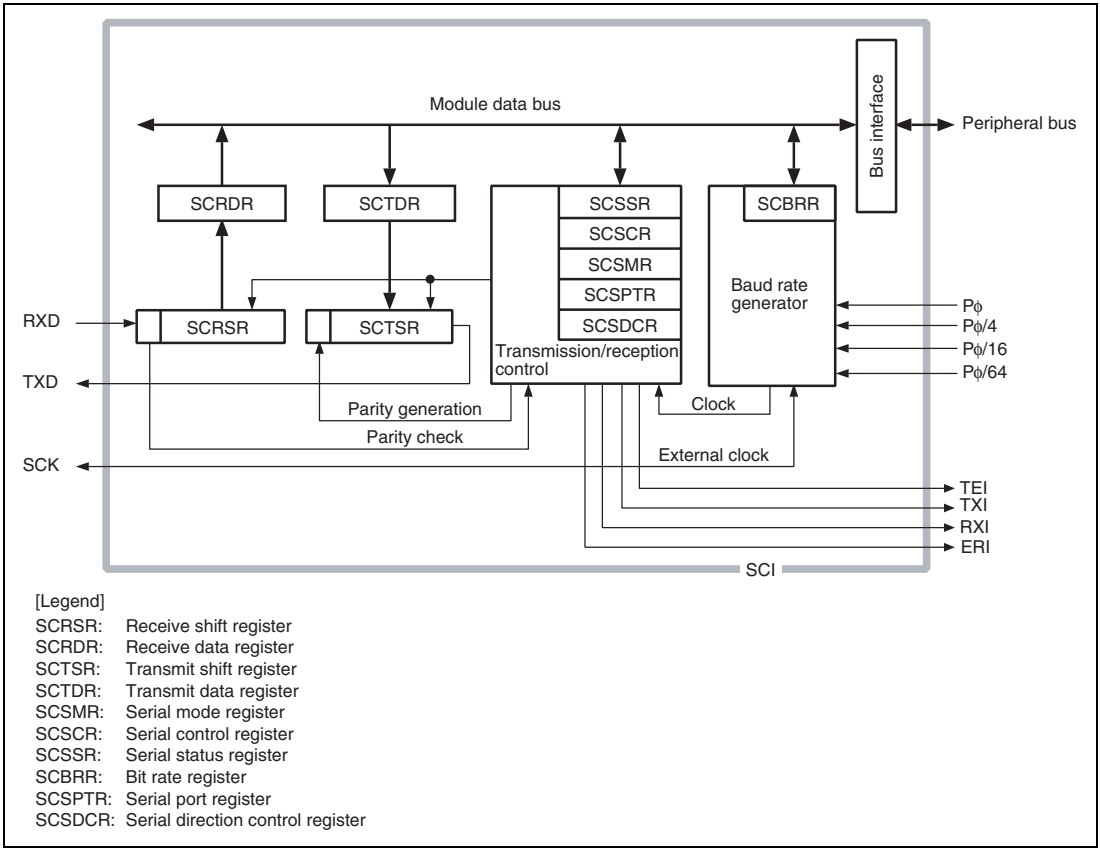


Figure 16.1 Block Diagram of SCI

16.2 Input/Output Pins

The SCI has the serial pins summarized in table 16.1.

Table 16.1 Pin Configuration

Channel	Pin Name*	I/O	Function
0	SCK0	I/O	SCI0 clock input/output
	RXD0	Input	SCI0 receive data input
	TXD0	Output	SCI0 transmit data output
1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
2	SCK2	I/O	SCI2 clock input/output
	RXD2	Input	SCI2 receive data input
	TXD2	Output	SCI2 transmit data output
4	SCK4	I/O	SCI4 clock input/output
	RXD4	Input	SCI4 receive data input
	TXD4	Output	SCI4 transmit data output

Note: * Pin names SCK, RXD, and TXD are used in the description for all channels, omitting the channel designation.

16.3 Register Descriptions

The SCI has the following registers for each channel. For details on register addresses and register states during each processing, refer to section 32, List of Registers.

Table 16.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'00	H'FFFF8000	8
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'FFFF8002	8
	Serial control register_0	SCSCR_0	R/W	H'00	H'FFFF8004	8
	Transmit data register_0	SCTDR_0	R/W	—	H'FFFF8006	8
	Serial status register_0	SCSSR_0	R/W	H'84	H'FFFF8008	8
	Receive data register_0	SCRDR_0	R	—	H'FFFF800A	8
	Serial direction control register_0	SCSDCR_0	R/W	H'F2	H'FFFF800C	8
	Serial port register_0	SCSPTR_0	R/W	H'0x	H'FFFF800E	8
1	Serial mode register_1	SCSMR_1	R/W	H'00	H'FFFF8800	8
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFF8802	8
	Serial control register_1	SCSCR_1	R/W	H'00	H'FFFF8804	8
	Transmit data register_1	SCTDR_1	R/W	—	H'FFFF8806	8
	Serial status register_1	SCSSR_1	R/W	H'84	H'FFFF8808	8
	Receive data register_1	SCRDR_1	R	—	H'FFFF880A	8
	Serial direction control register_1	SCSDCR_1	R/W	H'F2	H'FFFF880C	8
	Serial port register_1	SCSPTR_1	R/W	H'0x	H'FFFF880E	8
2	Serial mode register_2	SCSMR_2	R/W	H'00	H'FFFF9000	8
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'FFFF9002	8
	Serial control register_2	SCSCR_2	R/W	H'00	H'FFFF9004	8
	Transmit data register_2	SCTDR_2	R/W	—	H'FFFF9006	8
	Serial status register_2	SCSSR_2	R/W	H'84	H'FFFF9008	8
	Receive data register_2	SCRDR_2	R	—	H'FFFF900A	8
	Serial direction control register_2	SCSDCR_2	R/W	H'F2	H'FFFF900C	8
	Serial port register_2	SCSPTR_2	R/W	H'0x	H'FFFF900E	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	Serial mode register_4	SCSMR_4	R/W	H'00	H'FFFFA000	8
	Bit rate register_4	SCBRR_4	R/W	H'FF	H'FFFFA002	8
	Serial control register_4	SCSCR_4	R/W	H'00	H'FFFFA004	8
	Transmit data register_4	SCTDR_4	R/W	—	H'FFFFA006	8
	Serial status register_4	SCSSR_4	R/W	H'84	H'FFFFA008	8
	Receive data register_4	SCRDR_4	R	—	H'FFFFA00A	8
	Serial direction control register_4	SCSDCR_4	R/W	H'F2	H'FFFFA00C	8
	Serial port register_4	SCSPTR_4	R/W	H'0x	H'FFFFA00E	8

16.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RXD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to SCRDR. The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

16.3.2 Receive Data Register (SCRDR)

SCRDR is a register that stores serial receive data. After receiving one byte of serial data, the SCI transfers the received data from the receive shift register (SCRSR) into SCRDR for storage and completes operation. After that, SCRSR is ready to receive data.

Since SCRSR and SCRDR work as a double buffer in this way, data can be received continuously.

SCRDR is a read-only register and cannot be written to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

16.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCI loads transmit data from the transmit data register (SCTDR) into SCTSR, then transmits the data serially from the TXD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from SCTDR into SCTSR and starts transmitting again. If the TDRE flag in the serial status register (SCSSR) is set to 1, the SCI does not transfer data from SCTDR to SCTSR. The CPU cannot read or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

16.3.4 Transmit Data Register (SCTDR)

SCTDR is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCTDR into SCTSR and starts serial transmission. If the next transmit data has been written to SCTDR during serial transmission from SCTSR, the SCI can transmit data continuously. SCTDR can always be written or read to by the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.5 Serial Mode Register (SCSMR)

SCSMR is an 8-bit register that specifies the SCI serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR.

Bit:	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCI operates in asynchronous or clock synchronous mode. 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data in asynchronous mode. In the clock synchronous mode, the data length is always eight bits, regardless of the CHR setting. When 7-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted. 0: 8-bit data 1: 7-bit data
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.

Bit	Bit Name	Initial value	R/W	Description
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in clock synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity 1: Odd parity If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clock synchronous mode because no stop bits are added. 0: One stop bit*[1] 1: Two stop bits*[2] When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. Notes: 1. When transmitting, a single 1-bit is added at the end of each transmitted character. 2. When transmitting, two 1 bits are added at the end of each transmitted character.
2	MP	0	R/W	Multiprocessor Mode (only in asynchronous mode) Enables or disables multiprocessor mode. The PE and O/$\overline{E}$ bit settings are ignored in multiprocessor mode. 0: Multiprocessor mode disabled 1: Multiprocessor mode enabled

Bit	Bit Name	Initial value	R/W	Description
1, 0	CKS[1:0]	00	R/W	Clock Select 1 and 0 Select the internal clock source of the on-chip baud rate generator. Four clock sources are available; Pϕ, Pϕ/4, Pϕ/16, and Pϕ/64. For further information on the clock source, bit rate register settings, and baud rate, see section 16.3.10, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

16.3.6 Serial Control Register (SCSCR)

SCSCR is an 8-bit register that enables or disables SCI transmission/reception and interrupt requests and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables a transmit-data-empty interrupt (TXI) to be issued when the TDRE flag in the serial status register (SCSSR) is set to 1 after serial transmit data is sent from the transmit data register (SCTDR) to the transmit shift register (SCTSR). TXI can be canceled by clearing the TDRE flag to 0 after reading TDRE = 1 or by clearing the TIE bit to 0. 0: Transmit-data-empty interrupt request (TXI) is disabled 1: Transmit-data-empty interrupt request (TXI) is enabled
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables a receive-data-full interrupt (RXI) and a receive error interrupt (ERI) to be issued when the RDRF flag in SCSSR is set to 1 after the serial data received is transferred from the receive shift register (SCRSR) to the receive data register (SCRDR). RXI can be canceled by clearing the RDRF flag after reading RDRF =1. ERI can be canceled by clearing the FER, PER, or ORER flag to 0 after reading 1 from the flag. Both RXI and ERI can also be canceled by clearing the RIE bit to 0. 0: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled 1: Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled

Bit	Bit Name	Initial value	R/W	Description
5	TE	0	R/W	Transmit Enable Enables or disables the SCI serial transmitter. 0: Transmitter disabled*[1] 1: Transmitter enabled*[2] Notes: 1. The TDRE flag in SCSSR is fixed at 1. 2. Serial transmission starts after writing transmit data into SCTDR and clearing the TDRE flag in SCSSR to 0 while the transmitter is enabled. Select the transmit format in the serial mode register (SCSMR) before setting TE to 1.
4	RE	0	R/W	Receive Enable Enables or disables the SCI serial receiver. 0: Receiver disabled*[1] 1: Receiver enabled*[2] Notes: 1. Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, and ORER). These flags retain their previous values. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in clock synchronous mode. Select the receive format in SCSMR before setting RE to 1.
3	MPIE	0	R/W	Multiprocessor Interrupt Enable (only when MP = 1 in SCSMR in asynchronous mode) When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped and setting of the RDRF, FER, and ORER status flags in SCSSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared to 0 and normal receiving operation is resumed. For details, refer to section 16.4.4, Multiprocessor Communication Function.

Bit	Bit Name	Initial value	R/W	Description
2	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables a transmit end interrupt (TEI) to be issued when no valid transmit data is found in SCTDR during MSB data transmission. TEI can be canceled by clearing the TEND flag to 0 (by clearing the TDRE flag in SCSSR to 0 after reading TDRE = 1) or by clearing the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled 1: Transmit end interrupt request (TEI) is enabled
1, 0	CKE[1:0]	00	R/W	Clock Enable 1 and 0 Select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output or serial clock input. When selecting the clock output in clock synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1 and then set bits CKE1 and CKE0. For details on clock source selection, refer to table 16.14. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (The input signal is ignored.) 01: Internal clock, SCK pin used for clock output*[1] 10: External clock, SCK pin used for clock input*[2] 11: External clock, SCK pin used for clock input*[2] • Clock synchronous mode 00: Internal clock, SCK pin used for synchronous clock output 01: Internal clock, SCK pin used for synchronous clock output 10: External clock, SCK pin used for synchronous clock input 11: External clock, SCK pin used for synchronous clock input Notes: 1. The output clock frequency is 16 times the bit rate. 2. The input clock frequency is 16 times the bit rate.

16.3.7 Serial Status Register (SCSSR)

SCSSR is an 8-bit register that contains status flags to indicate the SCI operating state.

The CPU can always read and write to SCSSR, but cannot write 1 to status flags TDRE, RDRF, ORER, PER, and FER. These flags can be cleared to 0 only after 1 is read from the flags. The TEND flag is a read-only bit and cannot be modified.

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

Bit	Bit Name	Initial value	R/W	Description
7	TDRE	1	R/(W)*	Transmit Data Register Empty Indicates whether data has been transferred from the transmit data register (SCTDR) to the transmit shift register (SCTSR) and SCTDR has become ready to be written with next serial transmit data. 0: Indicates that SCTDR holds valid transmit data [Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and transmit data is transferred to SCTDR while the DISEL bit of MRB in the DTC is 0 (except when the DTC transfer counter value has become H'0000). 1: Indicates that SCTDR does not hold valid transmit data [Setting conditions] • By a power-on reset or in module standby mode • When the TE bit in SCSCR is 0 • When data is transferred from SCTDR to SCTSR and data can be written to SCTDR

Bit	Bit Name	Initial value	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates that the received data is stored in the receive data register (SCRDR). 0: Indicates that valid received data is not stored in SCRDR [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and data is transferred from SCRDR while the DISEL bit of MRB in the DTC is 0 (except when the DTC transfer counter value has become H'0000). 1: Indicates that valid received data is stored in SCRDR [Setting condition] • When serial reception ends normally and receive data is transferred from SCRSR to SCRDR Note: SCRDR and the RDRF flag are not affected and retain their previous states even if an error is detected during data reception or if the RE bit in the serial control register (SCSCR) is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the received data will be lost.

Bit	Bit Name	Initial value	R/W	Description
5	ORER	0	R/(W)*	Overrun Error Indicates that an overrun error occurred during reception, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*[1] [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to ORER after reading ORER = 1 1: Indicates that an overrun error occurred during reception*[2] [Setting condition] • When the next serial reception is completed while RDRF = 1 Notes: 1. The ORER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0. 2. The receive data prior to the overrun error is retained in SCRDR, and the data received subsequently is lost. Subsequent serial reception cannot be continued while the ORER flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
4	FER	0	R/(W)*	Framing Error Indicates that a framing error occurred during data reception in asynchronous mode, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*1 [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to FER after reading FER = 1 1: Indicates that a framing error occurred during reception [Setting condition] • When the SCI founds that the stop bit at the end of the received data is 0 after completing reception*2 Notes: 1. The FER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0. 2. In 2-stop-bit mode, only the first stop bit is checked for a value to 1; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to SCRDR but the RDRF flag is not set. Subsequent serial reception cannot be continued while the FER flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
3	PER	0	R/(W)*	Parity Error Indicates that a parity error occurred during data reception in asynchronous mode, causing abnormal termination. 0: Indicates that reception is in progress or was completed successfully*1 [Clearing conditions] • By a power-on reset or in module standby mode • When 0 is written to PER after reading PER = 1 1: Indicates that a parity error occurred during reception*2 [Setting condition] • When the number of 1s in the received data and parity does not match the even or odd parity specified by the O/$\overline{E}$ bit in the serial mode register (SCSMR). Notes: 1. The PER flag is not affected and retains its previous value when the RE bit in SCSCR is cleared to 0. 2. If a parity error occurs, the receive data is transferred to SCRDR but the RDRF flag is not set. Subsequent serial reception cannot be continued while the PER flag is set to 1.

Bit	Bit Name	Initial value	R/W	Description
2	TEND	1	R	Transmit End Indicates that no valid data was in SCTDR during transmission of the last bit of the transmit character and transmission has ended. The TEND flag is read-only and cannot be modified. 0: Indicates that transmission is in progress [Clearing condition] • When 0 is written to TDRE after reading TDRE = 1 1: Indicates that transmission has ended [Setting conditions] • By a power-on reset or in module standby mode • When the TE bit in SCSCR is 0 • When TDRE = 1 during transmission of the last bit of a 1-byte serial transmit character Note: The TEND flag value becomes undefined if data is written to SCTDR by activating the DTC by a TXI interrupt. In this case, do not use the TEND flag as the transmit end flag.
1	MPB	0	R	Multiprocessor Bit Stores the multiprocessor bit found in the receive data. When the RE bit in SCSCR is cleared to 0, its previous state is retained.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Specifies the multiprocessor bit value to be added to the transmit frame.

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

16.3.8 Serial Port Register (SCSPTR)

SCSPTR is an 8-bit register that controls input/output and data for the ports multiplexed with the SCI function pins. Data to be output through the TXD pin can be specified to control break of serial transfer. Through bits 3 and 2, data reading and writing through the SCK pin can be specified. Bit 7 enables or disables RXI interrupts. The CPU can always read and write to SCSPTR. When reading the value on the SCI pins, use the respective port register. For details, refer to section 23, I/O Ports.

Bit:	7	6	5	4	3	2	1	0
	EIO	-	-	-	SPB1IO	SPB1DT	-	SPB0DT
Initial value:	0	0	0	0	0	Undefined	0	1
R/W:	R/W	-	-	-	R/W	W	-	W

Bit	Bit Name	Initial value	R/W	Description
7	EIO	0	R/W	Error Interrupt Only Enables or disables RXI interrupts. While the EIO bit is set to 1, the SCI does not request an RXI interrupt to the CPU even if the RIE bit is set to 1. 0: The RIE bit enables or disables RXI and ERI interrupts. While the RIE bit is 1, RXI and ERI interrupts are sent to the INTC. 1: While the RIE bit is 1, only the ERI interrupt is sent to the INTC.
6 to 4	—	All 0	—	Reserved These bits are always read as 0. The write value should always be 0.
3	SPB1IO	0	R/W	Clock Port Input/Output in Serial Port Specifies the input/output direction of the SCK pin in the serial port. To output the data specified in the SPB1DT bit through the SCK pin as a port output pin, set the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in SCSCR to 0. 0: Does not output the SPB1DT bit value through the SCK pin. 1: Outputs the SPB1DT bit value through the SCK pin.

Bit	Bit Name	Initial value	R/W	Description
2	SPB1DT	Undefined	W	Clock Port Data in Serial Port Specifies the data output through the SCK pin in the serial port. Output should be enabled by the SPB1IO bit (for details, refer to the SPB1IO bit description). When output is enabled, the SPB1DT bit value is output through the SCK pin. 0: Low level is output 1: High level is output
1	—	0	—	Reserved This bit is always read as 0. The write value should always be 0.
0	SPB0DT	1	W	Serial Port Break Data Controls the TXD pin by the TE bit in SCSCR. However, TXD pin function should be selected by the pin function controller (PFC). This is a read-only bit. The read value is undefined.

TE bit setting in SCSCR	SPB0DT bit setting	TXD pin state
0	0	Low output
0	1	High output (initial state)
1	*	Transmit data output in accord with serial core logic.

Note: * Don't care

16.3.9 Serial Direction Control Register (SCSDCR)

The DIR bit in the serial direction control register (SCSDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	DIR	-	-	-
Initial value:	1	1	1	1	0	0	1	0
R/W:	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	DIR	0	R/W	Data Transfer Direction Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format. 0: SCTDR contents are transmitted in LSB-first order Receive data is stored in SCRDR in LSB-first 1: SCTDR contents are transmitted in MSB-first order Receive data is stored in SCRDR in MSB-first
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

16.3.10 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR.

The SCBRR setting is calculated as follows:

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Asynchronous mode:

- When the ABCS bit in serial extended mode register (SCSEMR) is 0

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- When the ABCS bit in serial extended mode register (SCSEMR) is 1

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clock synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting value should satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 16.3.)

Table 16.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

Note: The bit rate error in asynchronous is given by the following formula:

- When the ABCS bit in serial extended mode register (SCSEMR) is 0

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

- When the ABCS bit in serial extended mode register (SCSEMR) is 1

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

Tables 16.4 to 16.6 show examples of SCBRR settings in asynchronous mode, and tables 16.7 to 16.9 show examples of SCBRR settings in clock synchronous mode.

Table 16.4 Bit Rates and SCBRR Settings in Asynchronous Mode (1)

Bit Rate (bits/s)	Pϕ (MHz)																	
	10			12			14			16			18			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	-0.25	2	212	0.03	2	248	-0.17	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	32	-1.36	0	38	0.16	0	45	-0.93	0	51	0.16	0	58	-0.69	0	64	0.16
14400	0	21	-1.36	0	25	0.16	0	29	1.27	0	34	-0.79	0	38	0.16	0	42	0.94
19200	0	15	1.73	0	19	-2.34	0	22	-0.93	0	25	0.16	0	28	1.02	0	32	-1.36
28800	0	10	-1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	-2.34	0	21	-1.36
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	7	1.73	0	9	-2.34	0	10	3.57	0	12	0.16	0	14	-2.34	0	15	1.73

Table 16.5 Bit Rates and SCBRR Settings in Asynchronous Mode (2)

Bit Rate (bits/s)	Pϕ (MHz)																	
	22			24			26			28			30			32		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	97	-0.35	3	106	-0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	-0.54	3	77	0.16	3	84	-0.43	3	90	0.16	3	97	-0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	-0.54	2	77	0.16	2	84	-0.43	2	90	0.16	2	97	-0.35	2	103	0.16
1200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2400	1	71	-0.54	1	77	0.16	1	84	-0.43	1	90	0.16	1	97	-0.35	1	103	0.16
4800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9600	0	71	-0.54	0	77	0.16	0	84	-0.43	0	90	0.16	0	97	-0.35	0	103	0.16
14400	0	47	-0.54	0	51	0.16	0	55	0.76	0	60	-0.39	0	64	0.16	0	68	0.64
19200	0	35	-0.54	0	38	0.16	0	41	0.76	0	45	-0.93	0	48	-0.35	0	51	0.16
28800	0	23	-0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	-1.36	0	34	-0.79
31250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38400	0	17	-0.54	0	19	-2.34	0	20	0.76	0	22	-0.93	0	23	1.73	0	25	0.16

Table 16.6 Bit Rates and SCBRR Settings in Asynchronous Mode (3)

Bit Rate (bits/s)	Pϕ (MHz) 34 n	N	Error (%)	36 n	N	Error (%)	38 n	N	Error (%)	40 n	N	Error (%)	50 n	N	Error (%)
110	3	150	-0.05	3	159	-0.12	3	168	0.19	3	177	-0.25	3	221	-0.02
150	3	110	-0.29	3	116	0.16	3	123	-0.24	3	129	0.16	3	162	-0.15
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16	3	80	0.47
600	2	110	-0.29	2	116	0.16	2	123	-0.24	2	129	0.16	2	162	-0.15
1200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16	2	80	0.47
2400	1	110	-0.29	1	116	0.16	1	123	-0.24	1	129	0.16	1	162	-0.15
4800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16	1	80	0.47
9600	0	110	-0.29	0	116	0.16	0	123	-0.24	0	129	0.16	0	162	-0.15
14400	0	73	-0.29	0	77	0.16	0	81	0.57	0	86	-0.22	0	108	-0.45
19200	0	54	0.62	0	58	-0.69	0	61	-0.24	0	64	0.16	0	80	0.47
28800	0	36	-0.29	0	38	0.16	0	40	0.57	0	42	0.94	0	53	0.47
31250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00	0	49	0
38400	0	27	-1.18	0	28	1.02	0	30	-0.24	0	32	-1.36	0	40	-0.76

Table 16.7 Bit Rates and SCBRR Settings in Clock Synchronous Mode (1)

Bit Rate (bits/s)	Pϕ (MHz)											
	10		12		14		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1000	2	155	2	187	2	218	2	249	3	69	3	77
2500	1	249	2	74	2	87	2	99	2	112	2	124
5000	1	124	1	149	1	174	1	199	1	224	1	249
10000	0	249	1	74	1	87	1	99	1	112	1	124
25000	0	99	0	119	0	139	0	159	0	179	0	199
50000	0	49	0	59	0	69	0	79	0	89	0	99
100000	0	24	0	29	0	34	0	39	0	44	0	49
250000	0	9	0	11	0	13	0	15	0	17	0	19
500000	0	4	0	5	0	6	0	7	0	8	0	9
1000000	—	—	0	2	—	—	0	3	—	—	0	4
2500000	0	0*	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*

Table 16.8 Bit Rates and SCBRR Settings in Clock Synchronous Mode (2)

Bit Rate (bits/s)	Pϕ (MHz)											
	22		24		26		28		30		32	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1000	3	85	3	93	3	101	3	108	3	116	3	124
2500	2	137	2	149	2	162	2	174	2	187	2	199
5000	2	68	2	74	2	80	2	87	2	93	2	99
10000	1	137	1	149	1	162	1	174	1	187	1	199
25000	0	219	0	239	1	64	1	69	1	74	1	79
50000	0	109	0	119	0	129	0	139	0	149	0	159
100000	0	54	0	59	0	64	0	69	0	74	0	79
250000	0	21	0	23	0	25	0	27	0	29	0	31
500000	0	10	0	11	0	12	0	13	0	14	0	15
1000000	—	—	0	5	—	—	0	6	—	—	0	7
2500000	—	—	—	—	—	—	—	—	0	2	—	—
5000000	—	—	—	—	—	—	—	—	—	—	—	—

Table 16.9 Bit Rates and SCBRR Settings in Clock Synchronous Mode (3)

Bit Rate (bits/s)	Pϕ (MHz)									
	34		36		38		40		50	
	n	N	n	N	n	N	n	N	n	N
250										
500										
1000	3	132	3	140	3	147	3	155	3	194
2500	2	212	2	224	2	237	2	249	3	77
5000	2	105	2	112	2	118	2	124	2	155
10000	1	212	1	224	1	237	1	249	2	77
25000	1	84	1	89	1	94	1	99	1	124
50000	0	169	0	179	0	189	0	199	0	249
100000	0	84	0	89	0	94	0	99	0	124
250000	0	33	0	35	0	37	0	39	0	49
500000	0	16	0	17	0	18	0	19	0	24
1000000	—	—	0	8	—	—	0	9	—	—
2500000	—	—	—	—	—	—	0	3	0	4
5000000	—	—	—	—	—	—	0	1		

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

*: Continuous transmission/reception is disabled.

Note: Settings with an error of 1% or less are recommended.

Table 16.10 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Table 16.11 indicates the maximum bit rates in clock synchronous mode when the baud rate generator is used. Tables 16.12 and 16.13 list the maximum rates for external clock input.

Table 16.10 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	At Non-Continuous Transmission/Reception			At Continuous Transmission/Reception		
		Settings			Settings	
Pϕ (MHz)	Maximum Bit Rate (bits/s)	n	N	Maximum Bit Rate (bits/s)	n	N
10	312,500	0	0	156,250	0	1
12	375,000	0	0	187,500	0	1
14	437,500	0	0	218,750	0	1
16	500,000	0	0	250,000	0	1
18	562,500	0	0	281,250	0	1
20	625,000	0	0	312,500	0	1
22	687,500	0	0	343,750	0	1
24	750,000	0	0	375,000	0	1
26	812,500	0	0	406,250	0	1
28	875,000	0	0	437,500	0	1
30	937,500	0	0	468,750	0	1
32	1,000,000	0	0	500,000	0	1
34	1,062,500	0	0	531,250	0	1
36	1,125,000	0	0	562,500	0	1
38	1,187,500	0	0	593,750	0	1
40	1,250,000	0	0	625,000	0	1
50	1,562,500	0	0	781,250	0	1

Table 16.11 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Clock Synchronous Mode)

	At Non-Continuous Transmission/Reception			At Continuous Transmission/Reception		
	Maximum Bit Rate (bits/s)	Settings		Maximum Bit Rate (bits/s)	Settings	
Pϕ (MHz)		n	N		n	N
10	2,500,000	0	0	1,250,000	0	1
12	3,000,000	0	0	1,500,000	0	1
14	3,500,000	0	0	1,750,000	0	1
16	4,000,000	0	0	2,000,000	0	1
18	4,500,000	0	0	2,250,000	0	1
20	5,000,000	0	0	2,500,000	0	1
22	5,500,000	0	0	2,750,000	0	1
24	6,000,000	0	0	3,000,000	0	1
26	6,500,000	0	0	3,250,000	0	1
28	7,000,000	0	0	3,500,000	0	1
30	7,500,000	0	0	3,750,000	0	1
32	8,000,000	0	0	4,000,000	0	1
34	8,500,000	0	0	4,250,000	0	1
36	9,000,000	0	0	4,500,000	0	1
38	9,500,000	0	0	4,750,000	0	1
40	10,000,000	0	0	5,000,000	0	1
50	12,500,000	0	0	6,250,000	0	1

Table 16.12 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10	2.5000	156250
12	3.0000	187500
14	3.5000	218750
16	4.0000	250000
18	4.5000	281250
20	5.0000	312500
22	5.5000	343750
24	6.0000	375000
26	6.5000	406250
28	7.0000	437500
30	7.5000	468750
32	8.0000	500000
34	8.5000	531250
36	9.0000	562500
38	9.5000	593750
40	10.0000	625000
50	12.5000	781250

Table 16.13 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
22	3.6667	3666666.7
24	4.0000	4000000.0
26	4.3333	4333333.3
28	4.6667	4666666.7
30	5.0000	5000000.0
32	5.3333	5333333.3
34	5.6667	5666666.7
36	6.0000	6000000.0
38	6.3333	6333333.3
40	6.6667	6666666.7
50	8.3333	8333333.3

16.4 Operation

16.4.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a clock synchronous mode in which communication is synchronized with clock pulses.

Asynchronous or clock synchronous mode is selected and the transmit format is specified in the serial mode register (SCSMR) as shown in table 16.14. The SCI clock source is selected by the combination of the C/$\overline{A}$ bit in SCSMR and the CKE1 and CKE0 bits in the serial control register (SCSCR) as shown in table 16.15.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and breaks.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the clock supplied by the on-chip baud rate generator and can output a clock with a frequency 16 times the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clock Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors.
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCI operates on the input serial clock. The on-chip baud rate generator is not used.

Table 16.14 SCSMR Settings and SCI Communication Formats

SCSMR Settings					SCI Communication Format		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7-bit	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clock synchronous	8-bit	Not set	None

[Legend]

x: Don't care

Table 16.15 SCSMR and SCSCR Settings and SCI Clock Source Selection

SCSMR SCSCR Settings					
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCI does not use the SCK pin.
		1			Clock with a frequency 16 times the bit rate is output.
	1	0		External	Input a clock with frequency 16 times the bit rate.
		1			
1	0	0	Clock synchronous	Internal	Serial clock is output.
		1			
	1	0		External	Input the serial clock.
		1			

16.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes at the falling edge of the start bit. The SCI samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

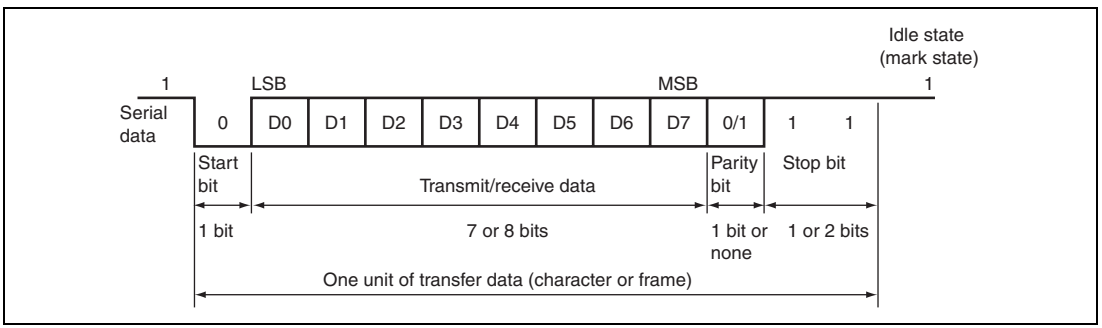


Figure 16.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 16.16 shows the transfer formats that can be selected in asynchronous mode. Any of 12 transfer formats can be selected according to the SCSMR settings.

Table 16.16 Serial Transfer Formats (Asynchronous Mode)

SCSMR Settings				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	x	1	0	S	8-bit data								MPB	STOP	
0	x	1	1	S	8-bit data								MPB	STOP	STOP
1	x	1	0	S	7-bit data							MPB	STOP		
1	x	1	1	S	7-bit data							MPB	STOP	STOP	

[Legend]
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit
x: Don't care

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE1 and CKE0 in the serial control register (SCSCR) (table 16.15).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to 16 times the desired bit rate.

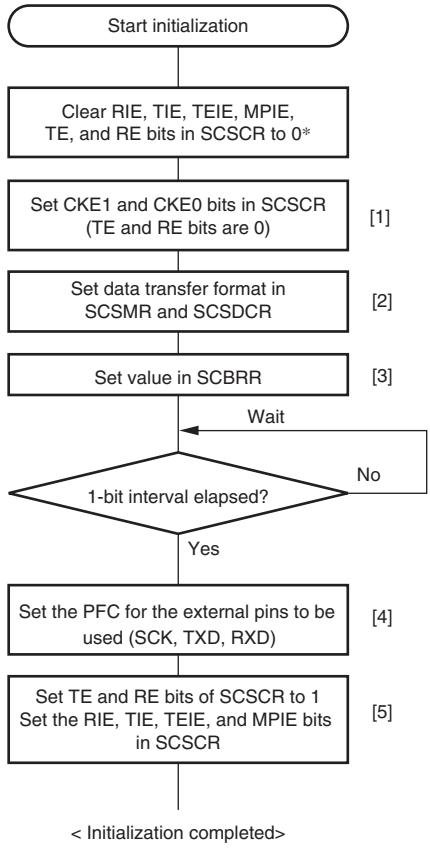
(3) Transmitting and Receiving Data

- SCI Initialization (Asynchronous Mode)

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI as follows.

When changing the operation mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing the TE bit to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR). Clearing the RE bit to 0, however, does not initialize the RDRF, PER, FER, and ORER flags or receive data register (SCRDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.



[1] Set the clock selection in SCSCR.

[2] Set the data transfer format in SCSMR and SCSDCR.

[3] Write a value corresponding to the bit rate to SCBRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RXD input during receiving and TXD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0. When CKE1 and CKE0 are 0 in asynchronous mode, setting the SCK pin is unnecessary. Outputting clocks from the SCK pin starts at synchronous clock output setting.

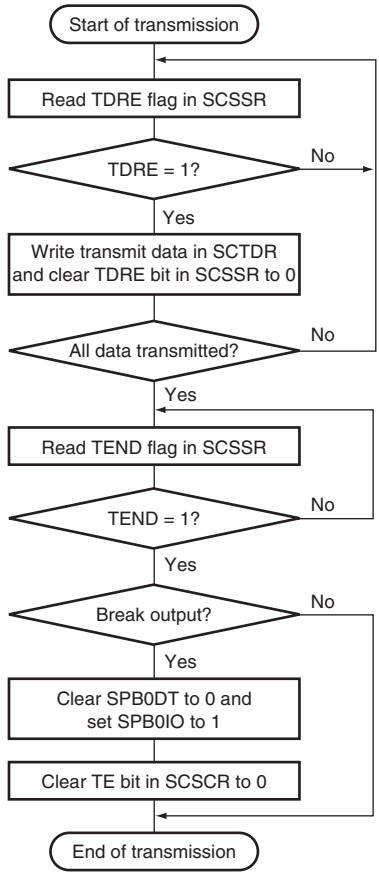
[5] Set the TE bit or RE bit in SCSCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting, and RXD pin is in an idle state for waiting the start bit during receiving.

Note : * In simultaneous transmit/receive operation, the TE and RE bits must be cleared to 0 or set to 1 simultaneously.

Figure 16.3 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Asynchronous Mode)

Figure 16.4 shows a sample flowchart for serial transmission. Use the following procedure for serial data transmission after enabling the SCI for transmission.



[1] SCI status check and transmit data write:

Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR, and clear the TDRE flag to 0.

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0.

When the DTC is activated by a transmit data empty interrupt (TXI) request to write data to SCTDR, clearing of the TDRE flag is automatic except when the transfer counter = 0 or DISEL = 1 as shown in the flowchart of DTC operation in section 8, Data Transfer Controller (DTC).

When the transfer counter = 0 or DISEL = 1, clear the TDRE flag in the interrupt handling routine.

[3] Break output at the end of serial transmission:

To output a break in serial transmission, clear the SPB0DT bit in SCSPTR to 0, then clear the TE bit in SCSCR to 0.

Figure 16.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in the serial status register (SCSSR). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR) and transfers the data from SCTDR to the transmit shift register (SCTSR).
2. After transferring data from SCTDR to SCTSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in the serial control register (SCSCR) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TXD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. (A format in which neither parity nor multiprocessor bit is output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCI checks the TDRE flag at the timing for sending the stop bit.
   If the TDRE flag is 0, the data is transferred from SCTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.
   If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the stop bit is sent, and then the "mark state" is entered in which 1 is output. If the TEIE bit in SCSCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 16.5 shows an example of the operation for transmission.

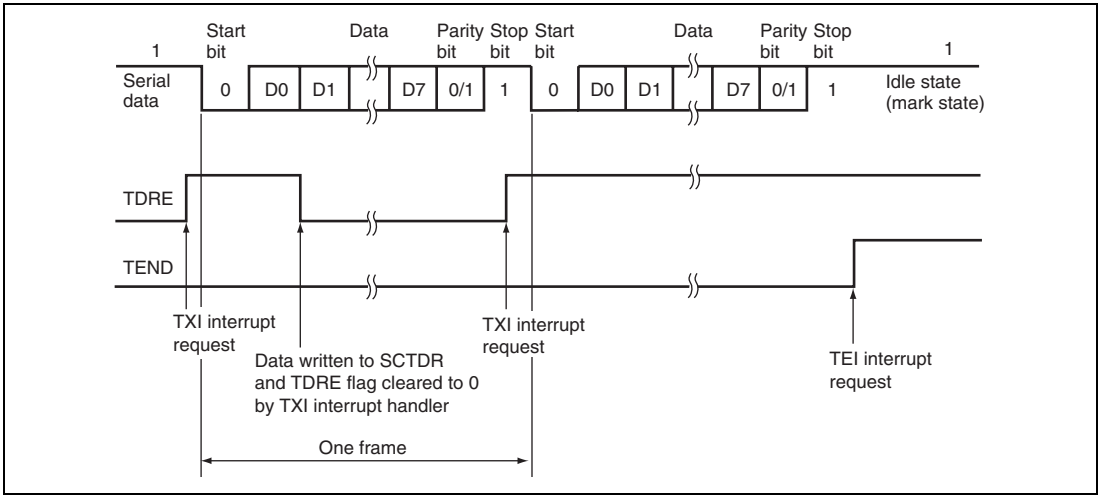


Figure 16.5 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

- Receiving Serial Data (Asynchronous Mode)

Figure 16.6 shows a sample flowchart for serial reception. Use the following procedure for serial data reception after enabling the SCI for reception.

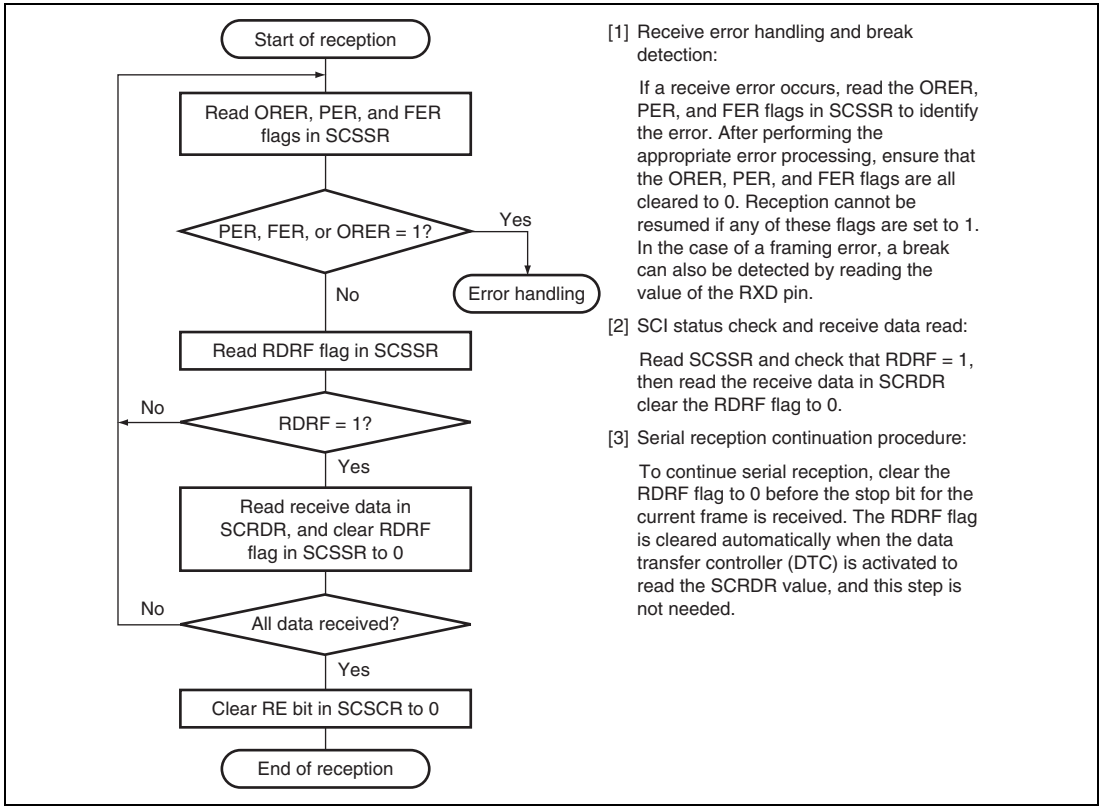


Figure 16.6 Sample Flowchart for Receiving Serial Data (1)

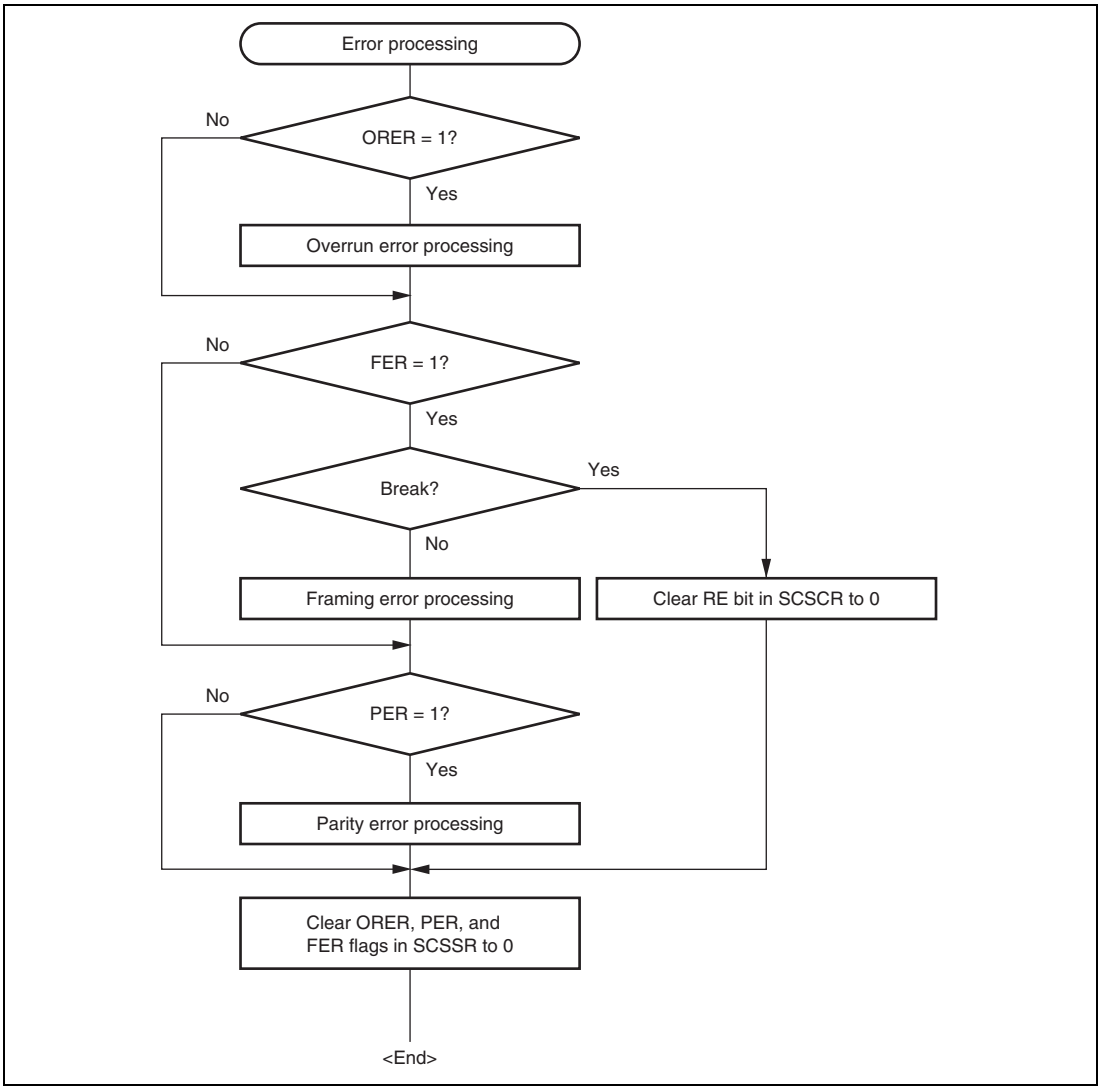


Figure 16.6 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCI operates as described below.

1. The SCI monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.
   After receiving these bits, the SCI carries out the following checks.
   A. Parity check: The SCI counts the number of 1s in the received data and checks whether the count matches the even or odd parity specified by the O/$\overline{E}$ bit in the serial mode register (SCSMR).
   B. Stop bit check: The SCI checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   C. Status check: The SCI checks whether the RDRF flag is 0 and the received data can be transferred from the receive shift register (SCRSR) to SCRDR.

   If all the above checks are passed, the RDRF flag is set to 1 and the received data is stored in SCRDR. If a receive error is detected, the SCI operates as shown in table 16.17.

   Note: When a receive error occurs, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the error flag to 0.

4. If the EIO bit in SCSPTR is cleared to 0 and the RIE bit in SCSCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated. If the RIE bit in SCSCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 16.17 Receive Errors and Error Conditions

Receive Error	Abbreviation	Error Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SCSSR is set to 1	The received data is not transferred from SCRSR to SCRDR.
Framing error	FER	When the stop bit is 0	The received data is transferred from SCRSR to SCRDR.
Parity error	PER	When the received data does not match the even or odd parity specified in SCSMR	The received data is transferred from SCRSR to SCRDR.

Figure 16.7 shows an example of the operation for reception.

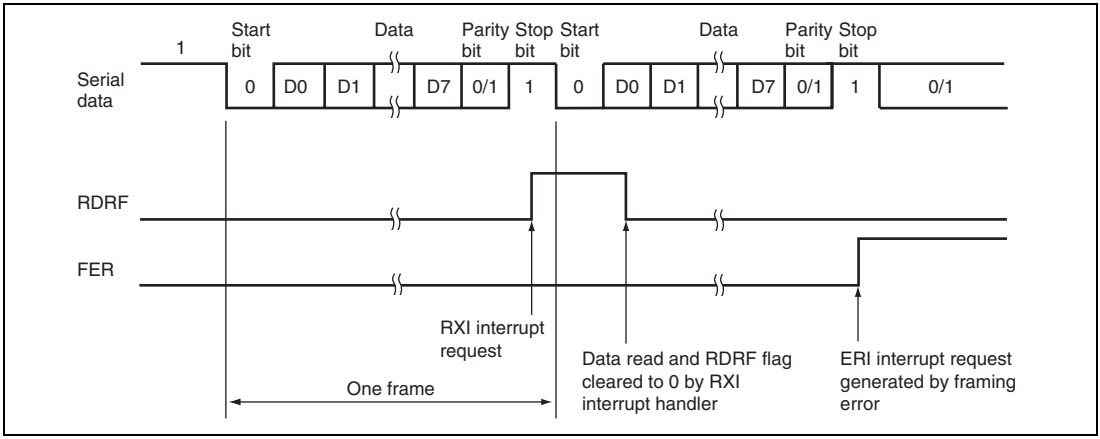


Figure 16.7 Example of SCI Receive Operation (8-Bit Data, Parity, One Stop Bit)

16.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. Both the transmitter and receiver have a double-buffered structure so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 16.8 shows the general format in clock synchronous serial communication.

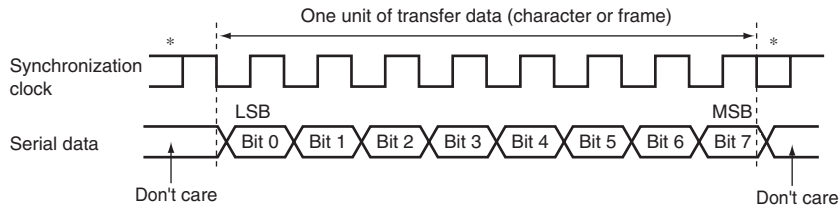


Figure 16.8 Data Format in Clock Synchronous Communication

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

In clock synchronous mode, the SCI transmits or receives data by synchronizing with the rising edge of the serial clock.

(1) Communication Format

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. For selection of the SCI clock source, see table 16.15.

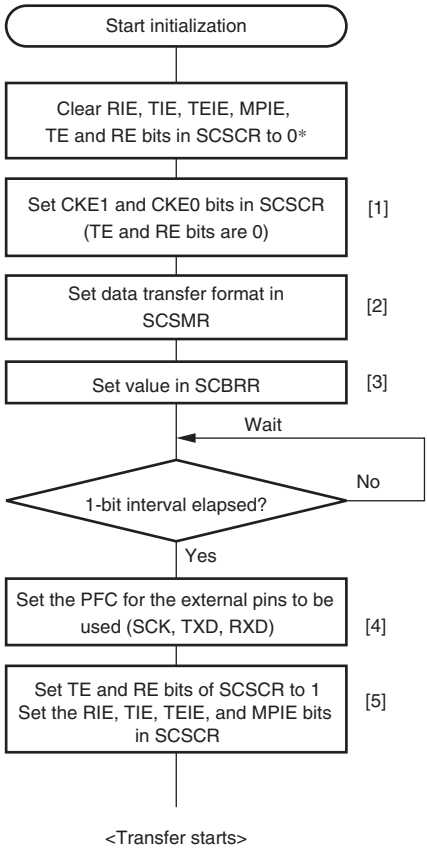
When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. However, in reception-only operation, the synchronizing clock is output until an overrun error occurs or the RE bit is cleared to 0. In operations for the reception of n characters, select the external clock as the clock source for the SCI. If the internal clock is to be used instead, set the RE and TE bits to 1, and then transmit n characters of dummy data during reception of the n characters to be received.

(3) Transmitting and Receiving Data

- SCI Initialization (Clock Synchronous Mode)

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCI. Clearing TE to 0 sets the TDRE flag to 1 and initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 16.9 shows a sample flowchart for initializing the SCI.



[1] Set the clock selection in SCSCR.

[2] Set the data transfer format in SCSMR.

[3] Write a value corresponding to the bit rate to SCBRR. Not necessary if an external clock is used.

[4] Set PFC of the external pin used. Set RXD input during receiving and TXD output during transmitting. Set SCK input/output according to contents set by CKE1 and CKE0.

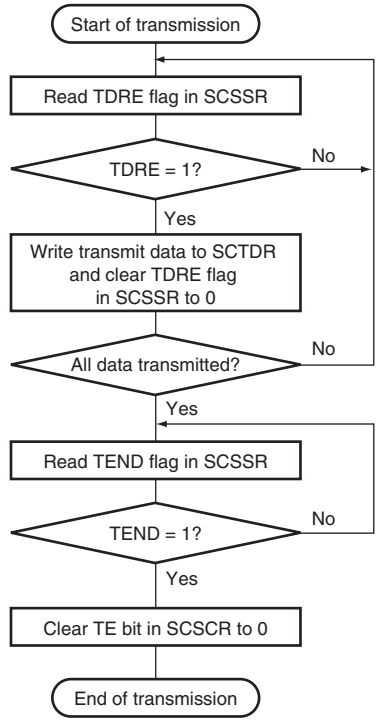
[5] Set the TE bit or RE bit in SCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting. When synchronous clock output (clock master) is set during receiving in clock synchronous mode, outputting clocks from the SCK pin starts.

Note: * In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 16.9 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Clock Synchronous Mode)

Figure 16.10 shows a sample flowchart for transmitting serial data. Use the following procedure for serial data transmission after enabling the SCI for transmission.



[1] SCI status check and transmit data write:

Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR, and clear the TDRE flag to 0.

[2] Serial transmission continuation procedure:

To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0.

When the DTC is activated by a transmit data empty interrupt (TXI) request to write data to SCTDR, clearing of the TDRE flag is automatic except when the transfer counter = 0 or DISEL = 1 as shown in the flowchart of DTC operation in section 8, Data Transfer Controller (DTC).

When the transfer counter = 0 or DISEL = 1, clear the TDRE flag in the interrupt handling routine.

Figure 16.10 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE flag in the serial status register (SCSSR). If it is cleared to 0, the SCI recognizes that data has been written to the transmit data register (SCTDR) and transfers the data from SCTDR to the transmit shift register (SCTSR).
2. After transferring data from SCTDR to SCTSR, the SCI sets the TDRE flag to 1 and starts transmission. If the transmit-data-empty interrupt enable bit (TIE) in the serial control register (SCSCR) is set to 1 at this time, a transmit-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCI outputs eight synchronous clock pulses. If an external clock source is selected, the SCI outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7). If the TDRE flag is 0, the data is transferred from SCTDR to SCTSR and serial transmission of the next frame is started, If the TDRE flag is 1, the TEND flag in SCSSR is set to 1, the MSB (bit 7) is sent, and then the TXD pin holds the states.
   If the TEIE bit in SCSCR is set to 1 at this time, a TEI interrupt request is generated.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 16.11 shows an example of SCI transmit operation.

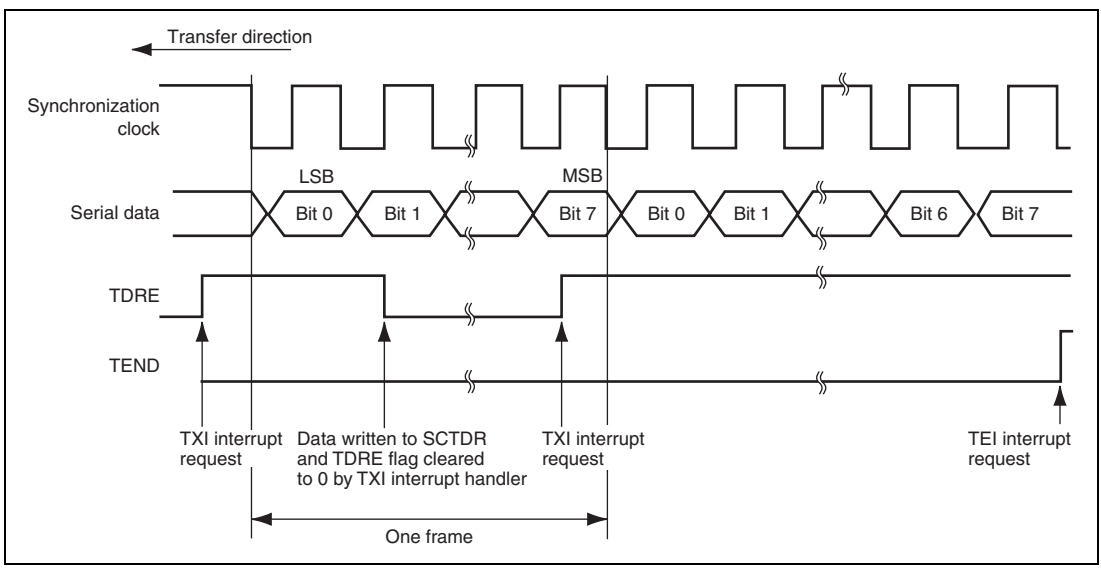
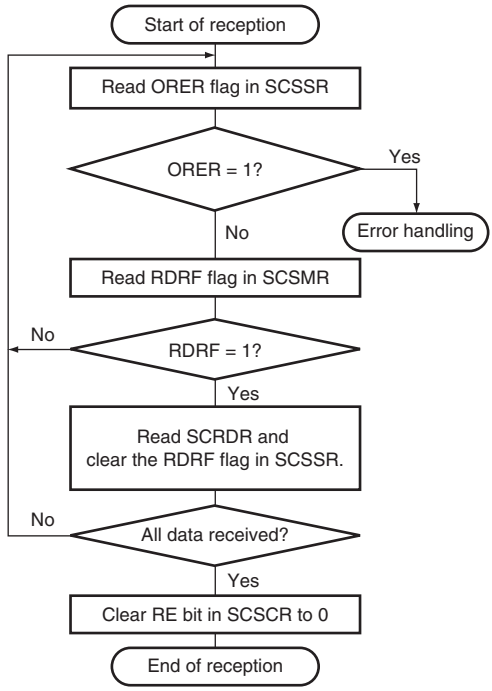


Figure 16.11 Example of SCI Transmit Operation

- Receiving Serial Data (Clock Synchronous Mode)

Figure 16.12 shows a sample flowchart for receiving serial data. Use the following procedure for serial data reception after enabling the SCIF for reception.

When switching from asynchronous mode to clock synchronous mode, make sure that the ORER, PER, and FER flags are all cleared to 0. If the FER or PER flag is set to 1, the RDRF flag will not be set and data reception cannot be started.



[1] Receive error handling:

Read the ORER flag in SCSSR to identify any error, perform the appropriate error handling, then clear the ORER flag to 0. Reception cannot be resumed while the ORER flag is set to 1.

[2] SCI status check and receive data read:

Read SCSSR and check that RDRF = 1, then read the receive data in SCRDR, and clear the RDRF flag to 0. The transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read the receive data register (SCRDR) and clear the RDRF flag to 0 before the MSB (bit 7) of the current frame is received. The RDRF flag is cleared automatically when the data transfer controller (DTC) is activated by a receive-data-full interrupt (RXI) request to read the SCRDR value, and this step is not needed.

Figure 16.12 Sample Flowchart for Receiving Serial Data (1)

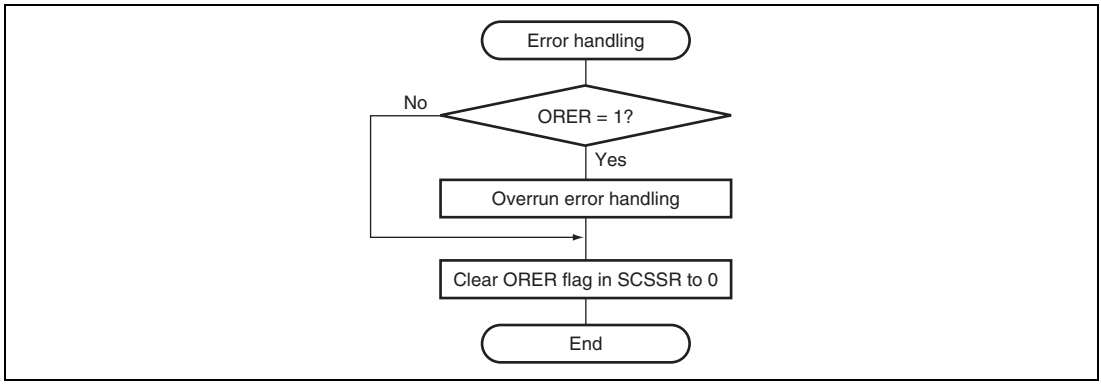


Figure 16.12 Sample Flowchart for Receiving Serial Data (2)

In receiving, the SCI operates as follows:

1. The SCI synchronizes with serial clock input or output and initializes internally.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from SCRSR to SCRDR. If this check is passed, the SCI sets the RDRF flag to 1 and stores the received data in SCRDR. If a receive error is detected, the SCI operates as shown in table 16.17. In this state, subsequent reception cannot be continued. In addition, the RDRF flag will not be set to 1 after reception; be sure to clear the RDRF flag to 0.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the RIE bit in SCSCR is also set to 1, the SCI requests a receive error interrupt (ERI).

Figure 16.13 shows an example of SCI receive operation.

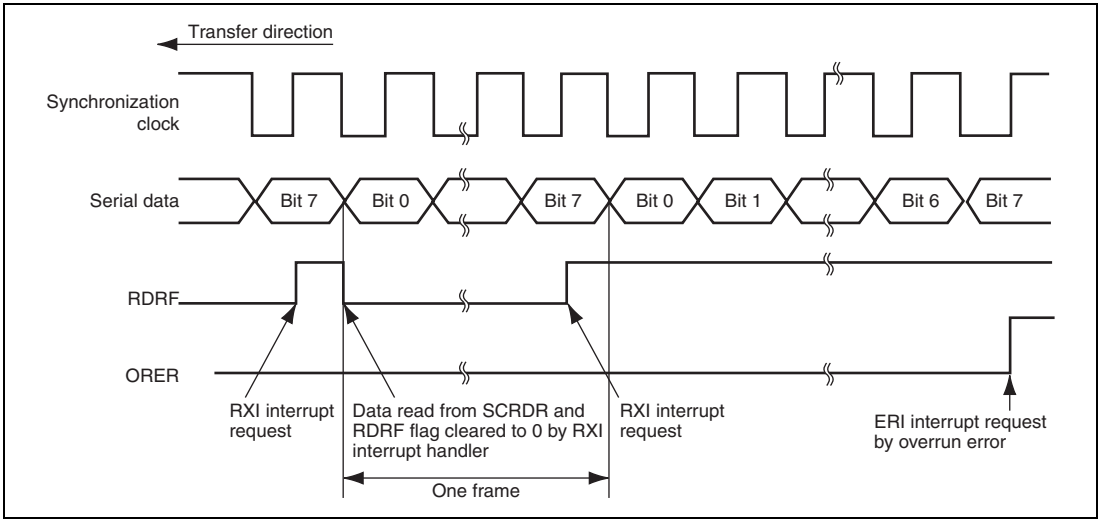
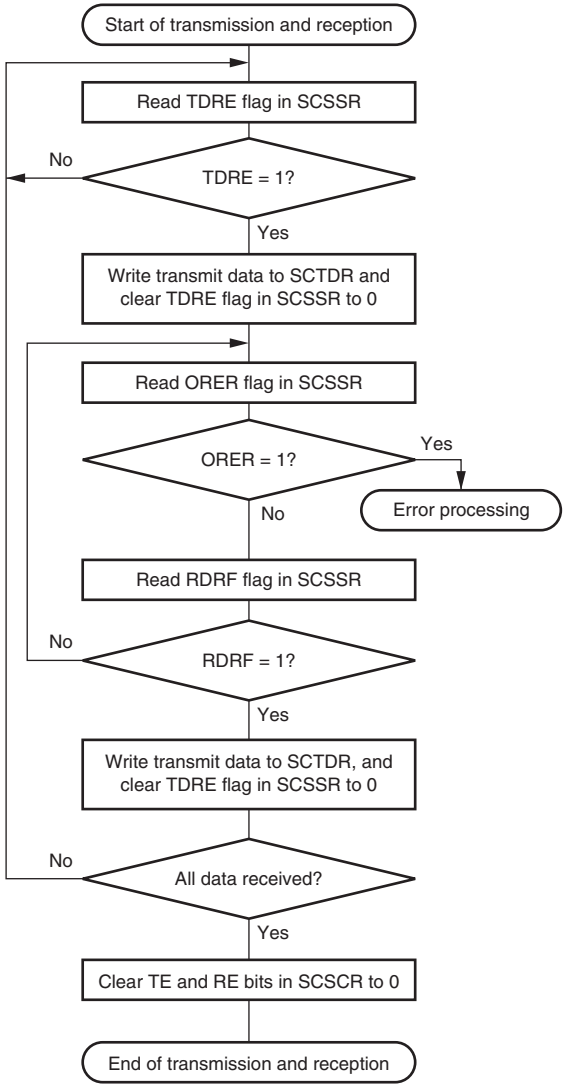


Figure 16.13 Example of SCI Receive Operation

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

Figure 16.14 shows a sample flowchart for transmitting and receiving serial data simultaneously. Use the following procedure for serial data transmission and reception after enabling the SCI for transmission and reception.



[1] SCI status check and transmit data write:
Read SCSSR and check that the TDRE flag is set to 1, then write transmit data to SCTDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[2] Receive error processing:
If a receive error occurs, read the ORER flag in SCSSR, and after performing the appropriate error processing, clear the ORER flag to 0.
Reception cannot be resumed if the ORER flag is set to 1.

[3] SCI status check and receive data read:
Read SCSSR and check that the RDRF flag is set to 1, then read the receive data in SCRDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[4] Serial transmission/reception continuation procedure:
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading SCRDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to SCTDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to SCTDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the SCRDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 16.14 Sample Flowchart for Transmitting/Receiving Serial Data

16.4.4 Multiprocessor Communication Function

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 16.15 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends the ID code of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added. The receiving station skips data until data with a 1 multiprocessor bit is sent. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCSCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from SCRSR to SCRDR, error flag detection, and setting the SCSSR status flags, RDRF, FER, and OER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SCSSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCSCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

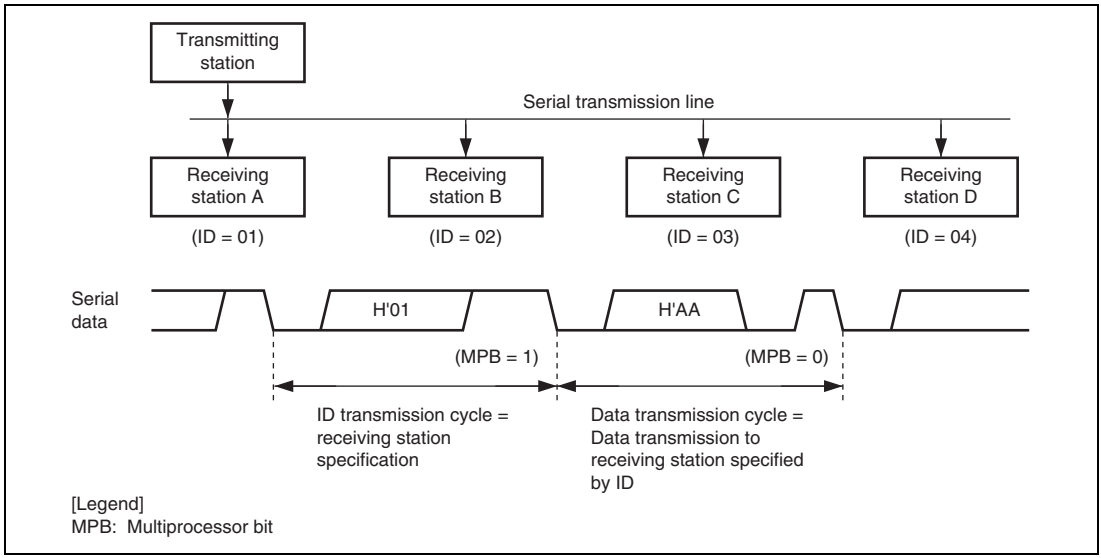
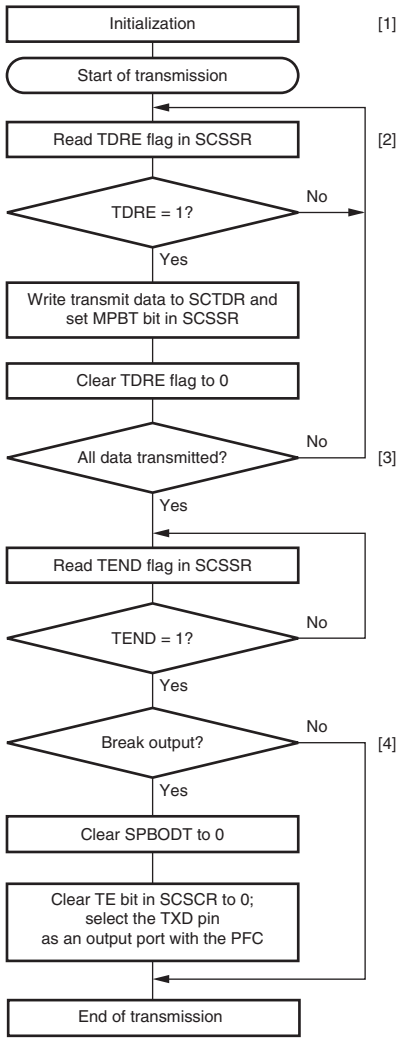


Figure 16.15 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

16.4.5 Multiprocessor Serial Data Transmission

Figure 16.16 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SCSSR to 1 before transmission. Keep MPBT at 1 until the ID is actually transmitted. For a data transmission cycle, clear the MPBT bit in SCSSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



[1] SCI initialization:
Set the TXD pin using the PFC. After the TE bit is set to 1, 1 is output for one frame, and transmission is enabled. However, data is not transmitted.

[2] SCI status check and transmit data write:
Read SCSSR and check that the TDRE flag is set to 1, then write data for transmission to SCTDR. Set the MPBT bit in SCSSR to 0 or 1. Finally, clear the TDRE flag to 0.
After initializing the SCI, when an ID is written to SCTDR register so as to transmit the ID, data is immediately transferred, and then the TDRE flag is set to 1. The MPBT bit must be held 1 because the ID is not transmitted from the TXD pin at this time. When the TDRE flag is set to 1 after data following the ID is written to SCTDR, clear the MPBT bit to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to SCTDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to SCTDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, first clear the SPBODT bit in the serial port register (SCSPTR) to 0, then clear the TE bit to 0 in SCSCR and use the PFC to

Figure 16.16 Sample Multiprocessor Serial Transmission Flowchart

16.4.6 Multiprocessor Serial Data Reception

Figure 16.18 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCSCR is set to 1, data is skipped until data with a 1 multiprocessor bit is sent. On receiving data with a 1 multiprocessor bit, the receive data is transferred to SCRDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 16.17 shows an example of SCI operation for multiprocessor format reception.

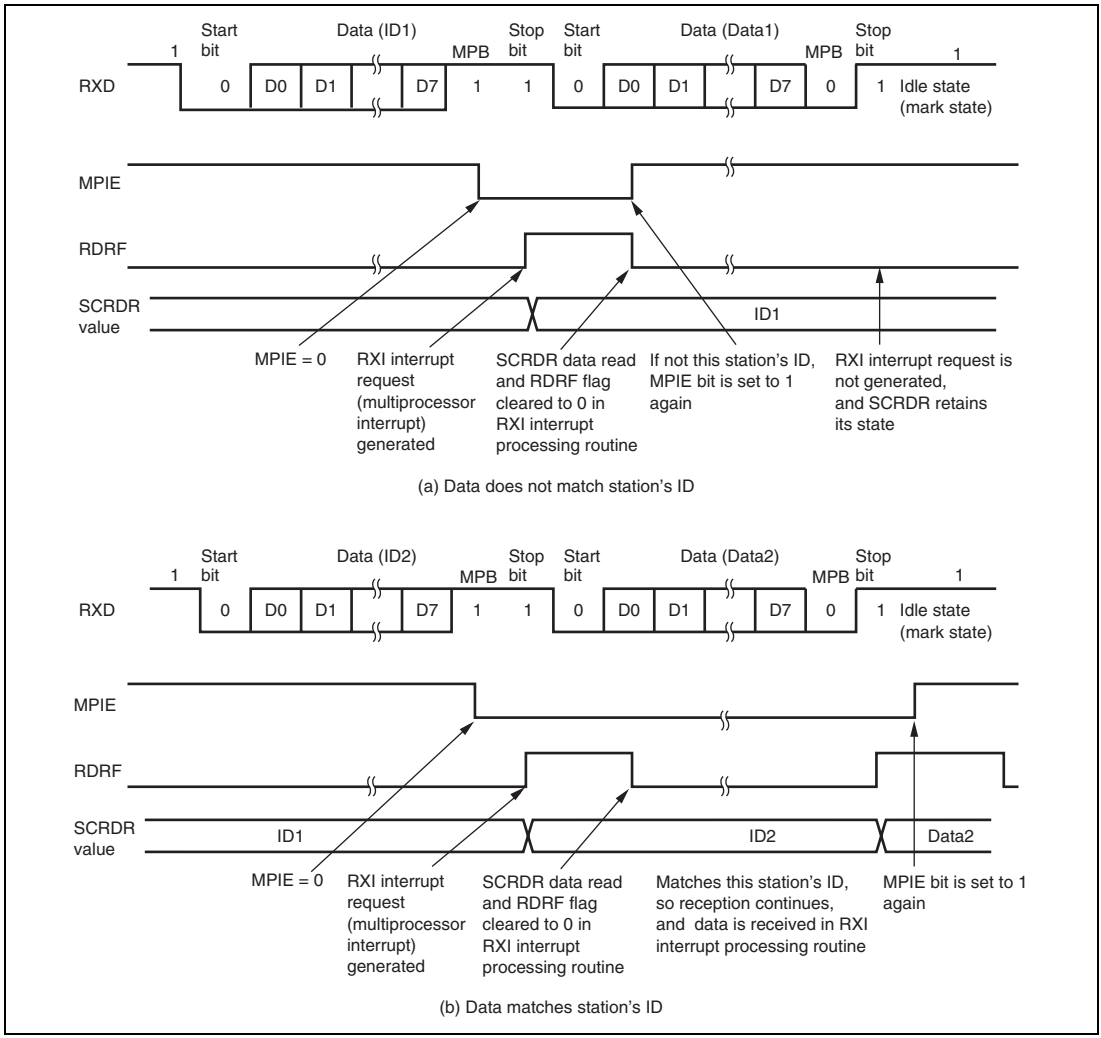
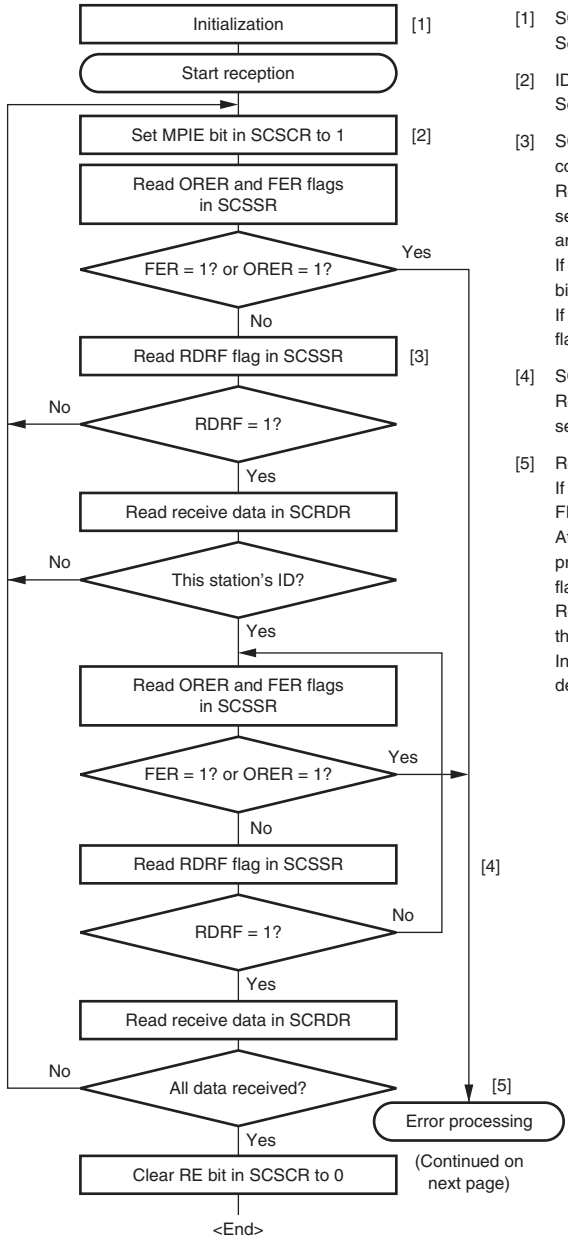


Figure 16.17 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[1] SCI initialization:
Set the RXD pin using the PFC.

[2] ID reception cycle:
Set the MPIE bit in SCSCR to 1.

[3] SCI status check, ID reception and comparison:
Read SCSSR and check that the RDRF flag is set to 1, then read the receive data in SCRDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SCSSR and check that the RDRF flag is set to 1, then read the data in SCRDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SCSSR to identify the error.
After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RXD pin value.

Figure 16.18 Sample Multiprocessor Serial Reception Flowchart (1)

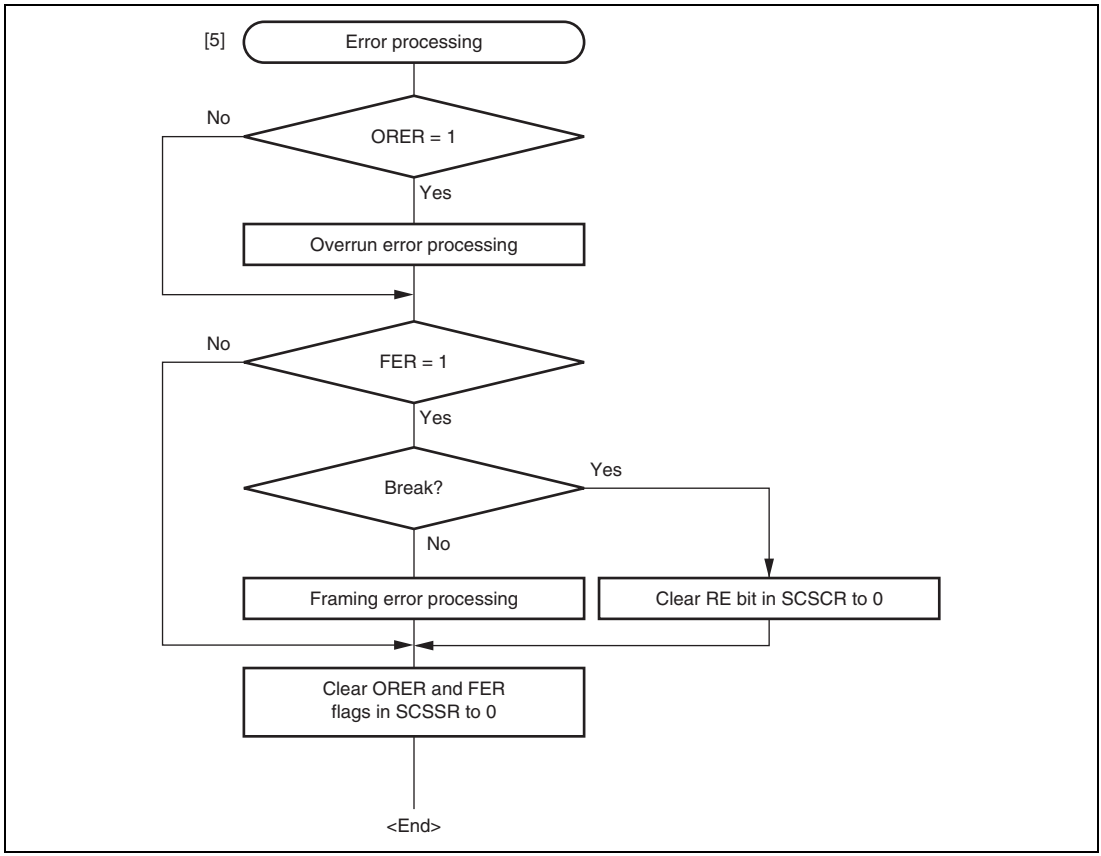


Figure 16.18 Sample Multiprocessor Serial Reception Flowchart (2)

16.5 SCI Interrupt Sources and DTC

The SCI has four interrupt sources: transmit end (TEI), receive error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI) interrupt requests.

Table 16.18 shows the interrupt sources. The interrupt sources are enabled or disabled by means of the TIE, RIE, and TEIE bits in SCSCR and the EIO bit in SCSPTR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When the TDRE flag in the serial status register (SCSSR) is set to 1, a TDR empty interrupt request is generated. This request can be used to activate the data transfer controller (DTC) to transfer data. The TDRE flag is automatically cleared to 0 when data is written to the transmit data register (SCTDR) through the DTC.

When the RDRF flag in SCSSR is set to 1, an RDR full interrupt request is generated. This request can be used to activate the DTC to transfer data. The RDRF flag is automatically cleared to 0 when data is read from the receive data register (SCRDR) through the DTC.

When the ORER, FER, or PER flag in SCSSR is set to 1, an ERI interrupt request is generated. This request cannot be used to activate the DTC. In processing for data reception, generation of ERI interrupt requests can only be enabled if generation of RXI interrupt requests is disabled. In this case, set the RIE bit and the EIO bit in SCSPTR to 1. However, note that the DMAC or DTC will not transfer received data since RXI interrupt requests are not generated while the EIO bit is set to 1.

When the TEND flag in SCSSR is set to 1, a TEI interrupt request is generated. This request cannot be used to activate the DTC.

The TXI interrupt indicates that transmit data can be written, and the TEI interrupt indicates that transmission has been completed.

Table 16.18 SCI Interrupt Sources

Interrupt Source	Description	DTC Activation
ERI	Interrupt caused by receive error (ORER, FER, or PER)	Not possible
RXI	Interrupt caused by receive data full (RDRF)	Possible
TXI	Interrupt caused by transmit data empty (TDRE)	Possible
TEI	Interrupt caused by transmit end (TENT)	Not possible

16.6 Serial Port Register (SCSPTR) and SCI Pins

The relationship between SCSPTR and the SCI pins is shown in figures 16.19 and 16.20.

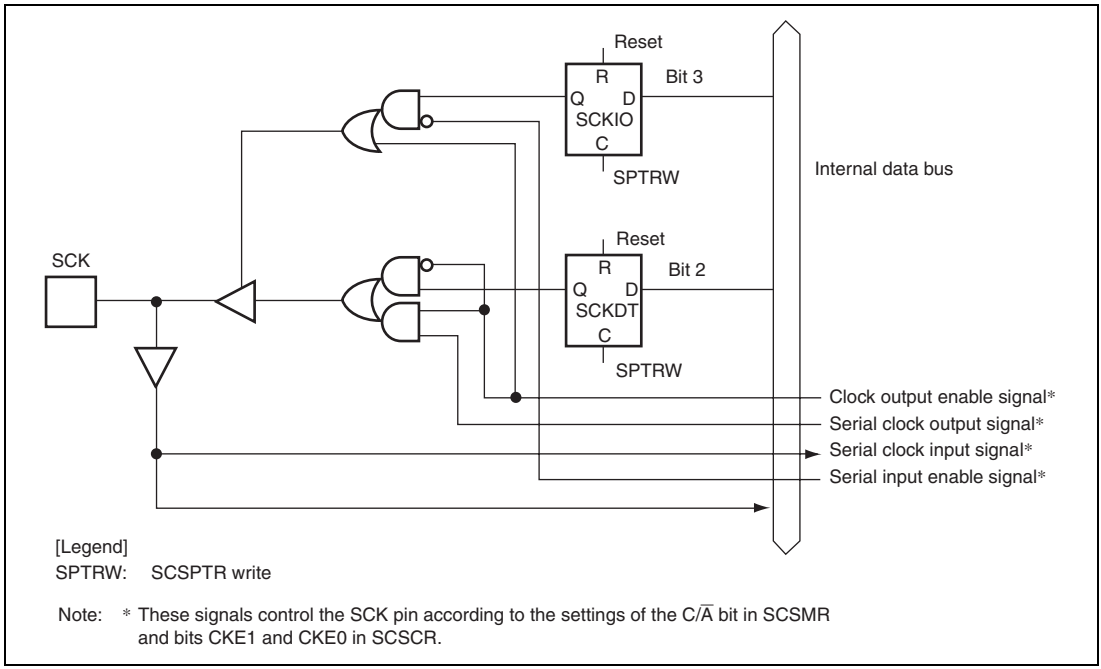


Figure 16.19 SCKIO Bit, SCKDT Bit, and SCK Pin

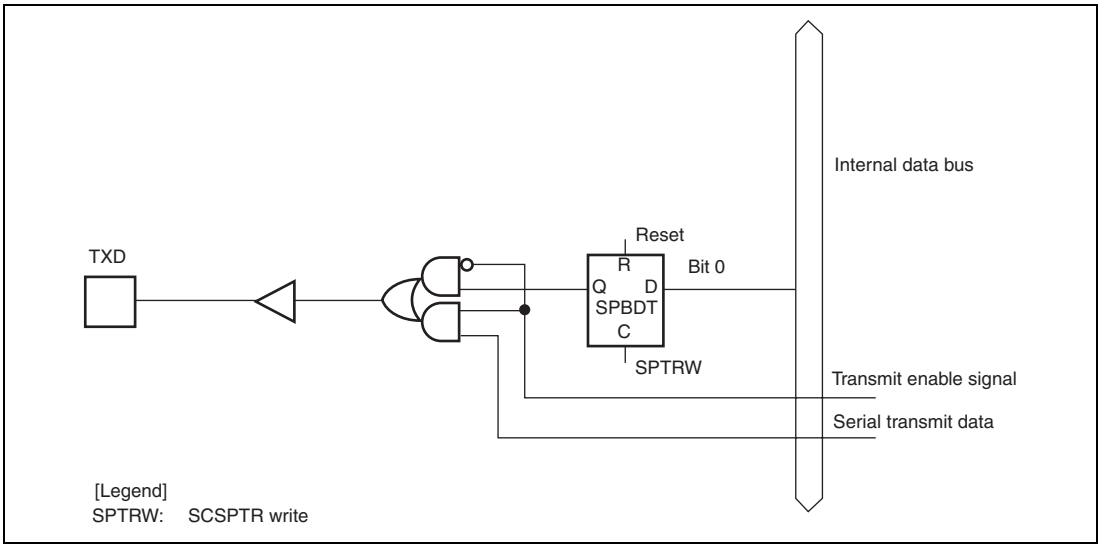


Figure 16.20 SPBDT Bit and TXD Pin

16.7 Usage Notes

16.7.1 SCTDR Writing and TDRE Flag

The TDRE flag in the serial status register (SCSSR) is a status flag indicating transferring of transmit data from SCTDR into SCTSR. The SCI sets the TDRE flag to 1 when it transfers data from SCTDR to SCTSR.

Data can be written to SCTDR regardless of the TDRE bit status.

If new data is written in SCTDR when TDRE is 0, however, the old data stored in SCTDR will be lost because the data has not yet been transferred to SCTSR. Before writing transmit data to SCTDR, be sure to check that the TDRE flag is set to 1.

16.7.2 Multiple Receive Error Occurrence

If multiple receive errors occur at the same time, the status flags in SCSSR are set as shown in table 16.19. When an overrun error occurs, data is not transferred from the receive shift register (SCRSR) to the receive data register (SCRDR) and the received data will be lost.

Table 16.19 SCSSR Status Flag Values and Transfer of Received Data

	SCSSR Status Flags				Receive Data Transfer from SCRSR to SCRDR
Receive Errors Generated	RDRF	ORER	FER	PER	
Overrun error	1	1	0	0	Not transferred
Framing error	0	0	1	0	Transferred
Parity error	0	0	0	1	Transferred
Overrun error + framing error	1	1	1	0	Not transferred
Overrun error + parity error	1	1	0	1	Not transferred
Framing error + parity error	0	0	1	1	Transferred
Overrun error + framing error + parity error	1	1	1	1	Not transferred

16.7.3 Break Detection and Processing

Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set. Note that, although transfer of receive data to SCRDR is halted in the break state, the SCI receiver continues to operate.

16.7.4 Sending a Break Signal

The I/O condition and level of the TXD pin are determined by SPB0DT bit in the serial port register (SCSPTR). This feature can be used to send a break signal.

Until TE bit is set to 1 (enabling transmission) after initializing, TXD pin does not work. During the period, mark status is performed by SPB0DT bit. Therefore, the SPB0DT bit should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB0DT bit to 0 (low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TXD pin.

16.7.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCI operates on a base clock with a frequency of 16 times the transfer rate in asynchronous mode. In reception, the SCI synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 16.21.

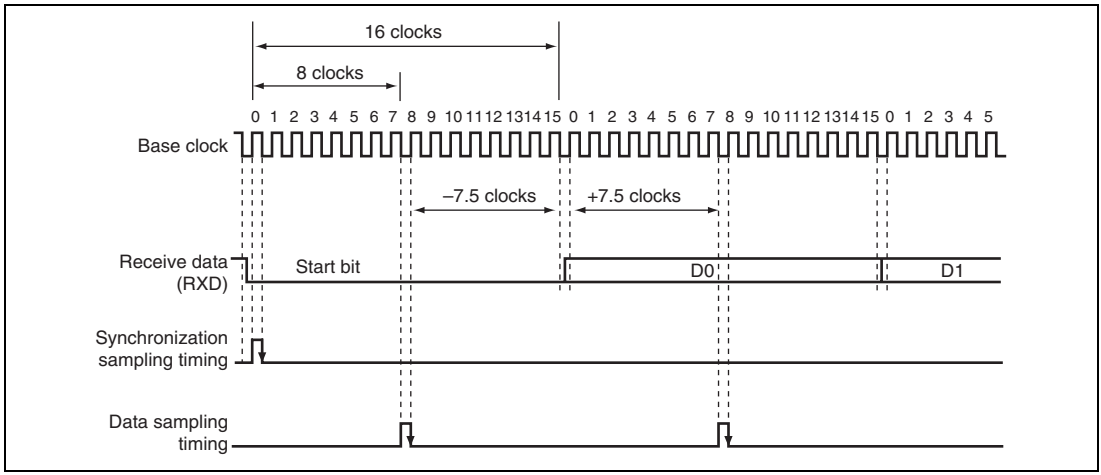


Figure 16.21 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1+F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$
$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

16.7.6 Note on Using DTC

When the external clock source is used for the clock for synchronization, input the external clock after waiting for five or more cycles of the peripheral operating clock after SCTDR is modified through the DTC. If a transmit clock is input within four cycles after SCTDR is modified, a malfunction may occur (figure 16.22).

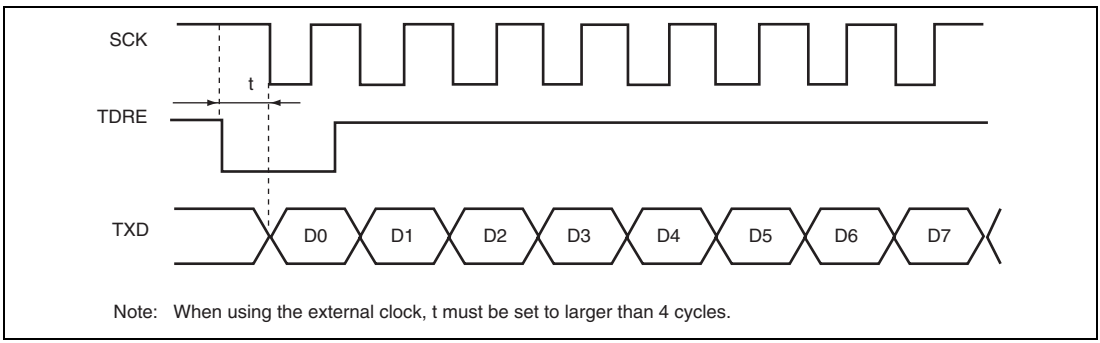


Figure 16.22 Example of Clock Synchronous Transfer Using DTC

When data is written to SCTDR by activating the DTC by a TXI interrupt, the TEND flag value becomes undefined. In this case, do not use the TEND flag as the transmit end flag.

16.7.7 Note on Using External Clock in Clock Synchronous Mode

TE and RE must be set to 1 after waiting for four or more cycles of the peripheral operating clock after the SCK external clock is changed from 0 to 1.

TE and RE must be set to 1 only while the SCK external clock is 1.

16.7.8 Module Standby Mode Setting

SCI operation can be disabled or enabled using the standby control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module standby mode. For details, refer to section 30, Power-Down Modes.

Section 17 Serial Communication Interface with FIFO (SCIF)

This LSI has one channel of serial communication interface with FIFO (SCIF) that supports both asynchronous and clocked synchronous serial communication. It also has 16-stage FIFO registers for both transmission and reception independently for each channel that enable this LSI to perform efficient high-speed continuous communication.

17.1 Features

- Asynchronous serial communication:
  - Serial data communication is performed by start-stop in character units. The SCIF can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communications interface adapter (ACIA), or any other communications chip that employs a standard asynchronous serial system. There are eight selectable serial data communication formats.
  - Data length: 7 or 8 bits
  - Stop bit length: 1 or 2 bits
  - Parity: Even, odd, or none
  - Receive error detection: Parity, framing, and overrun errors
  - Break detection: Break is detected when a framing error is followed by at least one frame at the space 0 level (low level). It is also detected by reading the RXD level directly from the serial port register when a framing error occurs.
- Clocked synchronous serial communication:
  - Serial data communication is synchronized with a clock signal. The SCIF can communicate with other chips having a clocked synchronous communication function. There is one serial data communication format.
  - Data length: 8 bits
  - Receive error detection: Overrun errors
- Full duplex communication: The transmitting and receiving sections are independent, so the SCIF can transmit and receive simultaneously. Both sections use 16-stage FIFO buffering, so high-speed continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates
- Internal or external transmit/receive clock source: From either baud rate generator (internal) or SCK pin (external)

- Four types of interrupts: Transmit-FIFO-data-empty interrupt, break interrupt, receive-FIFO-data-full interrupt, and receive-error interrupts are requested independently.
- When the SCIF is not in use, it can be stopped by halting the clock supplied to it, saving power.
- The quantity of data in the transmit and receive FIFO data registers and the number of receive errors of the receive data in the receive FIFO data register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.

Figure 17.1 shows a block diagram of the SCIF.

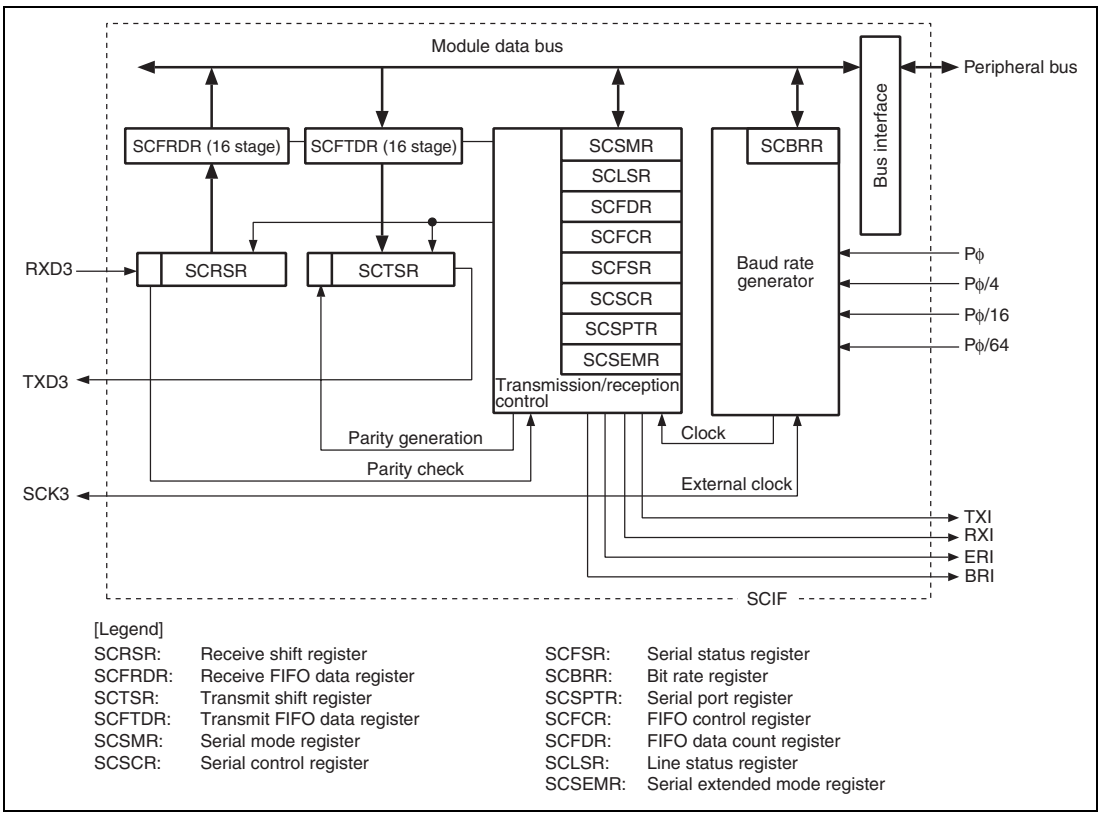


Figure 17.1 Block Diagram of SCIF

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the SCIF.

Table 17.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
3	Serial clock pins	SCK3	I/O	Clock I/O
	Receive data pins	RXD3	Input	Receive data input
	Transmit data pins	TXD3	Output	Transmit data output

17.3 Register Descriptions

The SCIF has the following registers.

Table 17.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
3	Serial mode register_3	SCSMR_3	R/W	H'0000	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	R/W	H'FF	H'FFFE9804	8
	Serial control register_3	SCSCR_3	R/W	H'0000	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	W	Undefined	H'FFFE980C	8
	Serial status register_3	SCFSR_3	R/(W)*1	H'0060	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	R	Undefined	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	R/W	H'0000	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	R	H'0000	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	R/W	H'005x	H'FFFE9820	16
	Line status register_3	SCLSR_3	R/(W)*2	H'0000	H'FFFE9824	16
	Serial extended mode register_3	SCSEMR_3	R/W	H'00	H'FFFE9900	8

Notes: 1. Only 0 can be written to clear the flag. Bits 15 to 8, 3, and 2 are read-only bits that cannot be modified.
2. Only 0 can be written to clear the flag. Bits 15 to 1 are read-only bits that cannot be modified.

17.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RXD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to the receive FIFO data register (SCFRDR).

The CPU cannot read or write to SCRSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

17.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to an undefined value by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

17.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The SCIF loads transmit data from the transmit FIFO data register (SCFTDR) into SCTSR, then transmits the data serially from the TXD pin, LSB (bit 0) first. After transmitting one data byte, the SCIF automatically loads the next transmit data from SCFTDR into SCTSR and starts transmitting again.

The CPU cannot read or write to SCTSR directly.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

17.3.4 Transmit FIFO Data Register (SCFTDR)

SCFTDR is a 16-byte FIFO register that stores data for serial transmission. When the SCIF detects that the transmit shift register (SCTSR) is empty, it moves transmit data written in the SCFTDR into SCTSR and starts serial transmission. Continuous serial transmission is performed until there is no transmit data left in SCFTDR. The CPU can write to SCFTDR at all times.

When SCFTDR is full of transmit data (16 bytes), no more data can be written. If writing of new data is attempted, the data is ignored.

SCFTDR is initialized to an undefined value by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

17.3.5 Serial Mode Register (SCSMR)

SCSMR specifies the SCIF serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to SCSMR. SCSMR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	C/$\overline{A}$	0	R/W	Communication Mode Selects whether the SCIF operates in asynchronous or clocked synchronous mode. 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length Selects 7-bit or 8-bit data length in asynchronous mode. In clocked synchronous mode, the data length is always 8 bits, regardless of the CHR setting. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.

Bit	Bit Name	Initial Value	R/W	Description
5	PE	0	R/W	Parity Enable Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In clocked synchronous mode, a parity bit is neither added nor checked, regardless of the PE setting. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode (O/$\overline{E}$) setting. Receive data parity is checked according to the even/odd (O/$\overline{E}$) mode setting.
4	O/$\overline{E}$	0	R/W	Parity mode Selects even or odd parity when parity bits are added and checked. The O/$\overline{E}$ setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The O/$\overline{E}$ setting is ignored in clocked synchronous mode, or in asynchronous mode when parity addition and checking is disabled. 0: Even parity*[1] 1: Odd parity*[2] Notes: 1. If even parity is selected, the parity bit is added to transmit data to make an even number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an even number of 1s in the received character and parity bit combined. 2. If odd parity is selected, the parity bit is added to transmit data to make an odd number of 1s in the transmitted character and parity bit combined. Receive data is checked to see if it has an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	Stop Bit Length Selects one or two bits as the stop bit length in asynchronous mode. This setting is used only in asynchronous mode. It is ignored in clocked synchronous mode because no stop bits are added. When receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit, but if the second stop bit is 0, it is treated as the start bit of the next incoming character. 0: One stop bit When transmitting, a single 1-bit is added at the end of each transmitted character. 1: Two stop bits When transmitting, two 1 bits are added at the end of each transmitted character.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKS[1:0]	00	R/W	Clock Select Select the internal clock source of the on-chip baud rate generator. For further information on the clock source, bit rate register settings, and baud rate, see section 17.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

17.3.6 Serial Control Register (SCSCR)

SCSCR operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR. SCSCR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to 1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled* Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmission trigger number to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive FIFO data full (RXI) interrupts requested when the RDF flag or DR flag in serial status register (SCFSR) is set to 1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to 1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to 1. 0: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled 1: Receive FIFO data full interrupt (RXI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled* Note: * RXI interrupt requests can be cleared by reading the DR or RDF flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0. ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0.
5	TE	0	R/W	Transmit Enable Enables or disables the serial transmitter. 0: Transmitter disabled 1: Transmitter enabled* Note: * Serial transmission starts after writing of transmit data into SCFTDR. Select the transmit format in SCSMR and SCFCR and reset the transmit FIFO before setting TE to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	RE	0	R/W	Receive Enable Enables or disables the serial receiver of the SCIF. 0: Receiver disabled*[1] 1: Receiver enabled*[2] Notes: 1. Clearing RE to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, PER, and ORER). These flags retain their previous values. 2. Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in clocked synchronous mode. Select the receive format in SCSMR and SCFCR and reset the receive FIFO before setting RE to 1.
3	REIE	0	R/W	Receive Error Interrupt Enable Enables or disables the receive-error (ERI) interrupts and break (BRI) interrupts. The setting of REIE bit is valid only when RIE bit is set to 0. 0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled 1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled* Note: * ERI or BRI interrupt requests can be cleared by reading the ER, BR or ORER flag after it has been set to 1, then clearing the flag to 0, or by clearing RIE and REIE to 0. Even if RIE is set to 0, when REIE is set to 1, ERI or BRI interrupt requests are enabled. Set so If SCIF wants to inform INTC of ERI or BRI interrupt requests during DMA transfer.

Bit	Bit Name	Initial Value	R/W	Description
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	CKE[1:0]	00	R/W	Clock Enable Select the SCIF clock source and enable or disable clock output from the SCK pin. Depending on CKE[1:0], the SCK pin can be used for serial clock output or serial clock input. If serial clock output is set in clocked synchronous mode, set the C/$\overline{A}$ bit in SCSMR to 1, and then set CKE[1:0]. • Asynchronous mode 00: Internal clock, SCK pin used for input pin (input signal is ignored) 01: Internal clock, SCK pin used for clock output (The output clock frequency is 16 times the bit rate.) 10: External clock, SCK pin used for clock input (The input clock frequency is 16 times the bit rate.) 11: Setting prohibited • Clocked synchronous mode 00: Internal clock, SCK pin used for serial clock output 01: Internal clock, SCK pin used for serial clock output 10: External clock, SCK pin used for serial clock input 11: Setting prohibited

17.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receive errors in the receive FIFO data register, and the lower 8 bits indicate the status flag indicating SCIF operating state.

The CPU can always read and write to SCFSR, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 3 (FER) and 2 (PER) are read-only bits that cannot be written.

When receive data in the receive FIFO data register is transferred by using the DTC/DMAC, the receive data is cleared in the receive FIFO data register. At the same time, the PER and FER bits in SCFSR are cleared. If the DTC/DMAC is used, an error is not judged by the FER or PER bit.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	PER[3:0]	0000	R	Number of Parity Errors Indicate the quantity of data including a parity error in the receive data stored in the receive FIFO data register (SCFRDR). The value indicated by bits 15 to 12 after the ER bit in SCFSR is set, represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER[3:0] shows 0000.
11 to 8	FER[3:0]	0000	R	Number of Framing Errors Indicate the quantity of data including a framing error in the receive data stored in SCFRDR. The value indicated by bits 11 to 8 after the ER bit in SCFSR is set, represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER[3:0] shows 0000.

Bit	Bit Name	Initial Value	R/W	Description
7	ER	0	R/(W)*	Receive Error Indicates the occurrence of a framing error, or of a parity error when receiving data that includes parity.*[1] 0: Receiving is in progress or has ended normally [Clearing conditions] • ER is cleared to 0 a power-on reset • ER is cleared to 0 when the chip is when 0 is written after 1 is read from ER 1: A framing error or parity error has occurred. [Setting conditions] • ER is set to 1 when the stop bit is 0 after checking whether or not the last stop bit of the received data is 1 at the end of one data receive operation*[2] • ER is set to 1 when the total number of 1s in the receive data plus parity bit does not match the even/odd parity specified by the O/$\overline{E}$ bit in SCSMR Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ER bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to SCFRDR and the receive operation is continued. Whether or not the data read from SCFRDR includes a receive error can be detected by the FER and PER bits in SCFSR. 2. In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.

Bit	Bit Name	Initial Value	R/W	Description
6	TEND	1	R/(W)*	Transmit End Indicates that when the last bit of a serial character was transmitted, SCFTDR did not contain valid data, so transmission has ended. 0: Transmission is in progress [Clearing condition] • TEND is cleared to 0 when 0 is written after 1 is read from TEND after transmit data is written in SCFTDR* 1: End of transmission [Setting conditions] • TEND is set to 1 when the chip is a power-on reset • TEND is set to 1 when TE is cleared to 0 in the serial control register (SCSCR) • TEND is set to 1 when SCFTDR does not contain receive data when the last bit of a one-byte serial character is transmitted Note: * Do not use this bit as a transmit end flag when the DMAC/DTC writes data to SCFTDR due to a TXI interrupt request.

Bit	Bit Name	Initial Value	R/W	Description
5	TDFE	1	R/(W)*	Transmit FIFO Data Empty Indicates that data has been transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), the quantity of data in SCFTDR has become less than the transmission trigger number specified by the TTRG1 and TTRG0 bits in the FIFO control register (SCFCR), and writing of transmit data to SCFTDR is enabled. 0: The quantity of transmit data written to SCFTDR is greater than the specified transmission trigger number [Clearing conditions] • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR after 1 is read from TDFE and then 0 is written • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR by the DMAC. • TDFE is cleared to 0 when data exceeding the specified transmission trigger number is written to SCFTDR by the DTC. (Except the transfer counter value of DTC has become H'0000) 1: The quantity of transmit data in SCFTDR is less than the specified transmission trigger number* [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the quantity of transmit data in SCFTDR becomes less than the specified transmission trigger number as a result of transmission. Note: * Since SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the specified transmission trigger number". If an attempt is made to write additional data, the data is ignored. The quantity of data in SCFTDR is indicated by the upper 8 bits of SCFDR.

Bit	Bit Name	Initial Value	R/W	Description
4	BRK	0	R/(W)*	Break Detection Indicates that a break signal has been detected in receive data. 0: No break signal received [Clearing conditions] • BRK is cleared to 0 when the chip is a power-on reset • BRK is cleared to 0 when software reads BRK after it has been set to 1, then writes 0 to BRK 1: Break signal received* [Setting condition] • BRK is set to 1 when data including a framing error is received, and a framing error occurs with space 0 in the subsequent receive data Note: * When a break is detected, transfer of the receive data (H'00) to SCFRDR stops after detection. When the break ends and the receive signal becomes mark 1, the transfer of receive data resumes.
3	FER	0	R	Framing Error Indication Indicates a framing error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive framing error occurred in the next data read from SCFRDR [Clearing conditions] • FER is cleared to 0 when the chip undergoes a power-on reset • FER is cleared to 0 when no framing error is present in the next data read from SCFRDR 1: A receive framing error occurred in the next data read from SCFRDR. [Setting condition] • FER is set to 1 when a framing error is present in the next data read from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
2	PER	0	R	Parity Error Indication Indicates a parity error in the data read from the next receive FIFO data register (SCFRDR) in asynchronous mode. 0: No receive parity error occurred in the next data read from SCFRDR [Clearing conditions] • PER is cleared to 0 when the chip undergoes a power-on reset • PER is cleared to 0 when no parity error is present in the next data read from SCFRDR 1: A receive parity error occurred in the next data read from SCFRDR [Setting condition] • PER is set to 1 when a parity error is present in the next data read from SCFRDR

Bit	Bit Name	Initial Value	R/W	Description
1	RDF	0	R/(W)*	Receive FIFO Data Full Indicates that receive data has been transferred to the receive FIFO data register (SCFRDR), and the quantity of data in SCFRDR has become more than the receive trigger number specified by the RTRG[1:0] bits in the FIFO control register (SCFCR). 0: The quantity of transmit data written to SCFRDR is less than the specified receive trigger number [Clearing conditions] • RDF is cleared to 0 by a power-on reset, standby mode • RDF is cleared to 0 when the SCFRDR is read until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number after 1 is read from RDF and then 0 is written • RDF is cleared to 0 when SCFRDR is read by the DMAC until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number. • RDF is cleared to 0 when SCFRDR is read by the DTC until the quantity of receive data in SCFRDR becomes less than the specified receive trigger number. (Except the transfer counter value of DTC has become H'0000) 1: The quantity of receive data in SCFRDR is more than the specified receive trigger number [Setting condition] • RDF is set to 1 when a quantity of receive data more than the specified receive trigger number is stored in SCFRDR* Note: * As SCFTDR is a 16-byte FIFO register, the maximum quantity of data that can be read when RDF is 1 becomes the specified receive trigger number. If an attempt is made to read after all the data in SCFRDR has been read, the data is undefined. The quantity of receive data in SCFRDR is indicated by the lower 8 bits of SCFDR.

Bit	Bit Name	Initial Value	R/W	Description
0	DR	0	R/(W)*	Receive Data Ready Indicates that the quantity of data in the receive FIFO data register (SCFRDR) is less than the specified receive trigger number, and that the next data has not yet been received after the elapse of 15 ETU from the last stop bit in asynchronous mode. In clocked synchronous mode, this bit is not set to 1. 0: Receiving is in progress, or no receive data remains in SCFRDR after receiving ended normally [Clearing conditions] • DR is cleared to 0 when the chip undergoes a power-on reset • DR is cleared to 0 when all receive data are read after 1 is read from DR and then 0 is written. • DR is cleared to 0 when all receive data in SCFRDR are read by the DMAC/DTC. 1: Next receive data has not been received [Setting condition] • DR is set to 1 when SCFRDR contains less data than the specified receive trigger number, and the next data has not yet been received after the elapse of 15 ETU from the last stop bit.* Note: * This is equivalent to 1.5 frames with the 8-bit, 1-stop-bit format. (ETU: elementary time unit)

Note: * Only 0 can be written to clear the flag after 1 is read.

17.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS[1:0] bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The SCBRR setting is calculated as follows:

Asynchronous mode:

- When the ABCS bit in serial extended mode register (SCSEMR) is 0

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

- When the ABCS bit in serial extended mode register (SCSEMR) is 1

$$N = \frac{P\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/s)
N: SCBRR setting for baud rate generator ($0 \leq N \leq 255$)
(The setting must satisfy the electrical characteristics.)
Pϕ: Operating frequency for peripheral modules (MHz)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (for the clock sources and values of n, see table 17.3.)

Table 17.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous is given by the following formula:

- When the ABCS bit in serial extended mode register (SCSEMR) is 0

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

- When the ABCS bit in serial extended mode register (SCSEMR) is 1

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 32 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 17.4 lists examples of SCBRR settings in asynchronous mode, and table 17.5 lists examples of SCBRR settings in clocked synchronous mode.

Table 17.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (1)

	Pϕ (MHz)																	
	10*			12*			14*			16*			18*			20		
Bit Rate (Bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	-0.25	2	212	0.03	2	248	-0.17	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1,200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2,400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4,800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9,600	0	32	-1.36	0	38	0.16	0	45	-0.93	0	51	0.16	0	58	-0.69	0	64	0.16
14,400	0	21	-1.36	0	25	0.16	0	29	1.27	0	34	-0.79	0	38	0.16	0	42	0.94
19,200	0	15	1.73	0	19	-2.34	0	22	-0.93	0	25	0.16	0	28	1.02	0	32	-1.36
28,800	0	10	-1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	-2.34	0	21	-1.36
31,250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38,400	0	7	1.73	0	9	-2.34	0	10	3.57	0	12	0.16	0	14	-2.34	0	15	1.73

Table 17.5 Bit Rates and SCBRR Settings (Asynchronous Mode) (2)

	Pϕ (MHz)																	
	22			24			26*			28*			30*			32*		
Bit Rate (Bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	97	-0.35	3	106	-0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	-0.54	3	77	0.16	3	84	-0.43	3	90	0.16	3	97	-0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	-0.54	2	77	0.16	2	84	-0.43	2	90	0.16	2	97	-0.35	2	103	0.16
1,200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2,400	1	71	-0.54	1	77	0.16	1	84	-0.43	1	90	0.16	1	97	-0.35	1	103	0.16
4,800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9,600	0	71	-0.54	0	77	0.16	0	84	-0.43	0	90	0.16	0	97	-0.35	0	103	0.16
14,400	0	47	-0.54	0	51	0.16	0	55	0.76	0	60	-0.39	0	64	0.16	0	68	0.64
19,200	0	35	-0.54	0	38	0.16	0	41	0.76	0	45	-0.93	0	48	-0.35	0	51	0.16
28,800	0	23	-0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	-1.36	0	34	-0.79
31,250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38,400	0	17	-0.54	0	19	-2.34	0	20	0.76	0	22	-0.93	0	23	1.73	0	25	0.16

Table 17.6 Bit Rates and SCBRR Settings (Asynchronous Mode) (3)

	Pϕ (MHz)														
	34*			36*			38*			40			50		
Bit Rate (Bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	150	-0.05	3	159	-0.12	3	168	-0.19	3	177	-0.25	3	221	-0.02
150	3	110	-0.29	3	116	0.16	3	123	-0.24	3	129	0.16	3	162	-0.15
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16	3	80	0.47
600	2	110	-0.29	2	116	0.16	2	123	-0.24	2	129	0.16	2	162	-0.15
1,200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16	2	80	0.47
2,400	1	110	-0.29	1	116	0.16	1	123	-0.24	1	129	0.16	1	162	-0.15
4,800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16	1	80	0.47
9,600	0	110	-0.29	0	116	0.16	0	123	-0.24	0	129	0.16	0	162	-0.15
14,400	0	73	-0.29	0	77	0.16	0	81	0.57	0	86	-0.22	0	108	-0.45
19,200	0	54	0.62	0	58	-0.69	0	61	-0.24	0	64	0.16	0	80	0.47
28,800	0	36	-0.29	0	38	0.16	0	40	0.57	0	42	0.94	0	53	0.47
31,250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00	0	49	0
38,400	0	27	-1.18	0	28	1.02	0	30	-0.24	0	32	-1.36	0	40	-0.76

Note: Cannot be set for this LSI.

* Settings with an error of 1% or less are recommended.

Table 17.7 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (1)

Bit Rate (Bit/s)	Pϕ (MHz)											
	10*1		12*1		14*1		16*1		18*1		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1,000	2	155	2	187	2	218	2	249	3	69	3	77
2,500	1	249	2	74	2	87	2	99	2	112	2	124
5,000	1	124	1	149	1	174	1	199	1	224	1	249
10,000	0	249	1	74	1	87	1	99	1	112	1	124
25,000	0	99	0	119	0	139	0	159	0	179	0	199
50,000	0	49	0	59	0	69	0	79	0	89	0	99
100,000	0	24	0	29	0	34	0	39	0	44	0	49
250,000	0	9	0	11	0	13	0	15	0	17	0	19
500,000	0	4	0	5	0	6	0	7	0	8	0	9
1,000,000	—	—	0	2	—	—	0	3	—	—	0	4
2,500,000	0	0*2	—	—	—	—	—	—	—	—	0	1
5,000,000			—	—	—	—	—	—	—	—	0	0*2

Table 17.8 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (2)

Bit Rate (Bit/s)	Pϕ (MHz)											
	22		24		26*1		28*1		30*1		32*1	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1,000	3	85	3	93	3	101	3	108	3	116	3	124
2,500	2	137	2	149	2	162	2	174	2	187	2	199
5,000	2	68	2	74	2	80	2	87	2	93	2	99
10,000	1	137	1	149	1	162	1	174	1	187	1	199
25,000	0	219	0	239	1	64	1	69	1	74	1	79
50,000	0	109	0	119	0	129	0	139	0	149	0	159
100,000	0	54	0	59	0	64	0	69	0	74	0	79
250,000	0	21	0	23	0	25	0	27	0	29	0	31
500,000	0	10	0	11	0	12	0	13	0	14	0	15
1000,000	—	—	0	5	—	—	0	6	—	—	0	7
2,500,000	—	—	—	—	—	—	—	—	0	2	—	—
5,000,000	—	—	—	—	—	—	—	—	—	—	—	—

Table 17.9 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (3)

Bit rate (Bits/s)	Pϕ (MHz)									
	34*[1]		36*[1]		38*[1]		40		50	
	n	N	n	N	n	N	n	N	n	N
250										
500										
1,000	3	132	3	140	3	147	3	155	3	194
2,500	2	212	2	224	2	237	2	249	3	77
5,000	2	105	2	112	2	118	2	124	2	155
10,000	1	212	1	224	1	237	1	249	2	77
25,000	1	84	1	89	1	94	1	99	1	124
50,000	0	169	0	179	0	189	0	199	0	249
100,000	0	84	0	89	0	94	0	99	0	124
250,000	0	33	0	35	0	37	0	39	0	49
500,000	0	16	0	17	0	18	0	19	0	24
1,000,000	—	—	0	8	—	—	0	9	—	—
2,500,000	—	—	—	—	—	—	0	3	0	4
5,000,000	—	—	—	—	—	—	0	1		

Notes: Settings with an error of 1% or less are recommended.

1. Cannot be set in this LSI.
2. Continuous transmission/reception is disabled.

[Legend]

Blank: Cannot be set.

—: Can be set with an error.

Table 17.10 indicates the maximum bit rates for various frequnecies in asynchronous mode when the baud rate generator is used. Table 17.11 indicates the maximum bit rates for various frequencies when the baud rate generator is used. Tables 17.12 and 17.13 list the maximum bit rates when the external clock input is used.

Table 17.10 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

	At Non-Continuous Transmission/Reception			At Continuous Transmission/Reception		
		Settings			Settings	
Pϕ (MHz)	Maximum Bit Rate (Bits/s)	n	N	Maximum Bit Rate (Bits/s)	n	N
10	312,500	0	0	156,250	0	1
12	375,000	0	0	187,500	0	1
14	437,500	0	0	218,750	0	1
16	500,000	0	0	250,000	0	1
18	562,500	0	0	281,250	0	1
20	625,000	0	0	312,500	0	1
22	687,500	0	0	343,750	0	1
24	750,000	0	0	375,000	0	1
26	812,500	0	0	406,250	0	1
28	875,000	0	0	437,500	0	1
30	937,500	0	0	468,750	0	1
32	1,000,000	0	0	500,000	0	1
34	1,062,500	0	0	531,250	0	1
36	1,125,000	0	0	562,500	0	1
38	1,187,500	0	0	593,750	0	1
40	1,250,000	0	0	625,000	0	1
50	1,562,500	0	0	781,250	0	1

Table 17.11 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Clocked Synchronous Mode)

	At Non-Continuous Transmission/Reception			At Continuous Transmission/Reception		
		Settings			Settings	
Pϕ (MHz)	Maximum Bit Rate (Bits/s)	n	N	Maximum Bit Rate (Bits/s)	n	N
10	2,500,000	0	0	1,250,000	0	1
12	3,000,000	0	0	1,500,000	0	1
14	3,500,000	0	0	1,750,000	0	1
16	4,000,000	0	0	2,000,000	0	1
18	4,500,000	0	0	2,250,000	0	1
20	5,000,000	0	0	2,500,000	0	1
22	5,500,000	0	0	2,750,000	0	1
24	6,000,000	0	0	3,000,000	0	1
26	6,500,000	0	0	3,250,000	0	1
28	7,000,000	0	0	3,500,000	0	1
30	7,500,000	0	0	3,750,000	0	1
32	8,000,000	0	0	4,000,000	0	1
34	8,500,000	0	0	4,250,000	0	1
36	9,000,000	0	0	4,500,000	0	1
38	9,500,000	0	0	4,750,000	0	1
40	10,000,000	0	0	5,000,000	0	1
50	12,500,000	0	0	6,250,000	0	1

Table 17.12 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	Maximum Bit Rate (Bits/s)	Maximum Bit Rate (Bits/s)
10*	2.5000	156,250
12*	3.0000	187,500
14*	3.5000	218,750
16*	4.0000	250,000
18*	4.5000	281,250
20	5.0000	312,500
22	5.5000	343,750
24	6.0000	375,000
26*	6.5000	406,250
28*	7.0000	437,500
30*	7.5000	468,750
32*	8.0000	500,000
34*	8.5000	531,250
36*	9.0000	562,500
38*	9.5000	593,750
40	10.0000	625,000
50	12.5000	781,250

Note: * Cannot be set in this LSI.

Table 17.13 Maximum Bit Rates with External Clock Input (Clocked Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Bits/s)
10*	1.6667	1,666,666.7
12*	2.0000	2,000,000.0
14*	2.3333	2,333,333.3
16*	2.6667	2,666,666.7
18*	3.0000	3,000,000.0
20	3.3333	3,333,333.3
22	3.6667	3,666,666.7
24	4.0000	4,000,000.0
26*	4.3333	4,333,333.3
28*	4.6667	4,666,666.7
30*	5.0000	5,000,000.0
32*	5.3333	5,333,333.3
34*	5.6667	5,666,666.7
36*	6.0000	6,000,000.0
38*	6.3333	6,333,333.3
40	6.6667	6,666,666.7
50	8.3333	8,333,333.3

Note: * Cannot be set in this LSI.

17.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive FIFO data registers, sets the trigger data quantity, and contains an enable bit for loop-back testing. SCFCR can always be read and written to by the CPU. It is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTRG[1:0]		TTRG[1:0]		-	TFRST	RFRST	LOOP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	RTRG[1:0]	00	R/W	Receive FIFO Data Trigger Set the quantity of receive data which sets the receive data full (RDF) flag in the serial status register (SCFSR). The RDF flag is set to 1 when the quantity of receive data stored in the receive FIFO register (SCFRDR) is increased more than the set trigger number shown below. • Asynchronous mode 00: 1 01: 4 10: 8 11: 14 • Clocked synchronous mode 00: 1 01: 2 10: 8 11: 14 Note: In clock synchronous mode, to transfer the receive data using DMAC, set the receive trigger number to 1. If set to other than 1, CPU must read the receive data left in SCFRDR.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	TTRG[1:0]	00	R/W	Transmit FIFO Data Trigger Set the quantity of remaining transmit data which sets the transmit FIFO data register empty (TDFE) flag in the serial status register (SCFSR). The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO data register (SCFTDR) becomes less than the set trigger number shown below. 00: 8 (8)* 01: 4 (12)* 10: 2 (14)* 11: 0 (16)* Note: * Values in parentheses mean the number of empty bytes in SCFTDR when the TDFE flag is set to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	TFRST	0	R/W	Transmit FIFO Data Register Reset Disables the transmit data in the transmit FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
1	RFRST	0	R/W	Receive FIFO Data Register Reset Disables the receive data in the receive FIFO data register and resets the data to the empty state. 0: Reset operation disabled* 1: Reset operation enabled Note: * Reset operation is executed by a power-on reset.
0	LOOP	0	R/W	Loop-Back Test Internally connects the transmit output pin (TXD) and receive input pin (RXD) and internally connects the $\overline{\text{RTS}}$ pin and $\overline{\text{CTS}}$ pin and enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

17.3.10 FIFO Data Count Register (SCFDR)

SCFDR is a 16-bit register which indicates the quantity of data stored in the transmit FIFO data register (SCFTDR) and the receive FIFO data register (SCFRDR).

It indicates the quantity of transmit data in SCFTDR with the upper 8 bits, and the quantity of receive data in SCFRDR with the lower 8 bits. SCFDR can always be read by the CPU. SCFDR is initialized to H'0000 by a power on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]					-	-	-	R[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	T[4:0]	00000	R	T4 to T0 bits indicate the quantity of non-transmitted data stored in SCFTDR. H'00 means no transmit data, and H'10 means that SCFTDR is full of transmit data.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	R[4:0]	00000	R	R4 to R0 bits indicate the quantity of receive data stored in SCFRDR. H'00 means no receive data, and H'10 means that SCFRDR full of receive data.

17.3.11 Serial Port Register (SCSPTR)

SCSPTR controls input/output and data of pins multiplexed to SCIF function. Bits 3 and 2 can control input/output data of SCK pin. Bits 1 and 0 can input data from RXD pin and output data to TXD pin, so they control break of serial transmitting/receiving.

The CPU can always read and write to SCSPTR. SCSPTR is initialized to H'0050 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	SCKIO	SCKDT	SPB2IO	SPB2DT
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	Undefined	0	Undefined
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	W	R/W	W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	SCKIO	0	R/W	SCK Port Input/Output Indicates input or output of the serial port SCK pin. When the SCK pin is actually used as a port outputting the SCKDT bit value, the CKE[1:0] bits in SCSCR should be cleared to 0. 0: SCKDT bit value not output to SCK pin 1: SCKDT bit value output to SCK pin

Bit	Bit Name	Initial Value	R/W	Description
2	SCKDT	Undefined	W	SCK Port Data Indicates the input/output data of the serial port SCK pin. Input/output is specified by the SCKIO bit. For output, the SCKDT bit value is output to the SCK pin. The SCK pin status is read from the SCKDT bit regardless of the SCKIO bit setting. However, SCK input/output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level
1	SPB2IO	0	R/W	Serial Port Break Input/Output Indicates input or output of the serial port TXD pin. When the TXD pin is actually used as a port outputting the SPB2DT bit value, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value not output to TXD pin 1: SPB2DT bit value output to TXD pin
0	SPB2DT	Undefined	W	Serial Port Break Data Indicates the input data of the RXD pin and the output data of the TXD pin used as serial ports. Input/output is specified by the SPB2IO bit. When the TXD pin is set to output, the SPB2DT bit value is output to the TXD pin. The RXD pin status is read from the SPB2DT bit regardless of the SPB2IO bit setting. However, RXD input and TXD output must be set in the PFC. 0: Input/output data is low level 1: Input/output data is high level

17.3.12 Line Status Register (SCLSR)

The CPU can always read or write to SCLSR, but cannot write 1 to the ORER flag. This flag can be cleared to 0 only if it has first been read (after being set to 1).

SCLSR is initialized to H'0000 by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ORER	0	R/(W)*	Overrun Error Indicates the occurrence of an overrun error. 0: Receiving is in progress or has ended normally*1 [Clearing conditions] • ORER is cleared to 0 when the chip is a power-on reset • ORER is cleared to 0 when 0 is written after 1 is read from ORER. 1: An overrun error has occurred*2 [Setting condition] • ORER is set to 1 when the next serial receiving is finished while the receive FIFO is full of 16-byte receive data. Notes: 1. Clearing the RE bit to 0 in SCSCR does not affect the ORER bit, which retains its previous value. 2. The receive FIFO data register (SCFRDR) retains the data before an overrun error has occurred, and the next received data is discarded. When the ORER bit is set to 1, the SCIF cannot continue the next serial reception.

17.3.13 Serial Extended Mode Register (SCSEMR)

SCSEMR is an 8-bit register that extends the SCIF functions. The transfer rate can be doubled by setting the basic clock in asynchronous mode.

Be sure to set this register to H'00 in clocked synchronous mode. SCSEMR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ABCS	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ABCS	0	R/W	Asynchronous Basic Clock Select Selects the basic clock for 1-bit period in asynchronous mode. Setting of ABCS is valid when the asynchronous mode bit (C/$\overline{A}$ in SCSMR) = 0. 0: Basic clock with a frequency of 16 times the transfer rate 1: Basic clock with a frequency of 8 times the transfer rate
6 to 0	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.

17.4 Operation

17.4.1 Overview

For serial communication, the SCIF has an asynchronous mode in which characters are synchronized individually, and a clocked synchronous mode in which communication is synchronized with clock pulses.

The SCIF has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU, and enabling continuous high-speed communication.

The transmission format is selected in the serial mode register (SCSMR), as shown in table 17.14. The SCIF clock source is selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCSCR), as shown in table 17.15.

(1) Asynchronous Mode

- Data length is selectable: 7 or 8 bits
- Parity bit is selectable. So is the stop bit length (1 or 2 bits). The combination of the preceding selections constitutes the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The number of stored data bytes is indicated for both the transmit and receive FIFO registers.
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of on-chip baud rate generator.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

(2) Clocked Synchronous Mode

- The transmission/reception format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIF clock source.
  - When an internal clock is selected, the SCIF operates using the clock of the on-chip baud rate generator, and outputs this clock to external devices as the synchronous clock.
  - When an external clock is selected, the SCIF operates on the input synchronous clock not using the on-chip baud rate generator.

Table 17.14 SCSMR Settings and SCIF Communication Formats

SCSMR					SCIF Communication Format		
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Parity Bit	Stop Bit Length
0	0	0	0	Asynchronous	8 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
	1	0	0		7 bits	Not set	1 bit
			1				2 bits
		1	0			Set	1 bit
			1				2 bits
1	x	x	x	Clocked synchronous	8 bits	Not set	None

[Legend]

x: Don't care

Table 17.15 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR				
Bit 7	Bit 1	Bit 0		Clock	
C/$\overline{A}$	CKE1	CKE0	Mode	Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin
		1			Outputs a clock with a frequency 16 times the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
		1		Setting prohibited	
1	0	x	Clocked synchronous	Internal	Outputs the serial clock
	1	0		External	Inputs the serial clock
		1		Setting prohibited	

[Legend]

x: Don't care

17.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCIF are independent, so full duplex communication is possible. The transmitter and receiver are 16-byte FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 17.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIF monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIF synchronizes at the falling edge of the start bit. The SCIF samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. Receive data is latched at the center of each bit.

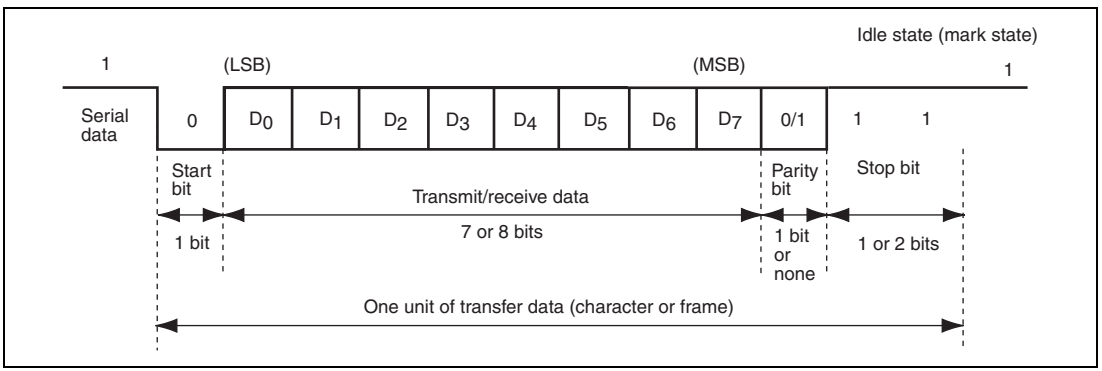


Figure 17.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 17.16 lists the eight communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SCSMR).

Table 17.16 Serial Communication Formats (Asynchronous Mode)

SCSMR Bits			Serial Transmit/Receive Format and Frame Length											
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	START	8-bit data								STOP		
0	0	1	START	8-bit data								STOP	STOP	
0	1	0	START	8-bit data								P	STOP	
0	1	1	START	8-bit data								P	STOP	STOP
1	0	0	START	7-bit data							STOP			
1	0	1	START	7-bit data							STOP	STOP		
1	1	0	START	7-bit data							P	STOP		
1	1	1	START	7-bit data							P	STOP	STOP	

[Legend]
START: Start bit
STOP: Stop bit
P: Parity bit

(2) Clock

An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock. The clock source is selected by the C/$\overline{A}$ bit in the serial mode register (SCSMR) and bits CKE[1:0] in the serial control register (SCSCR). For clock source selection, refer to table 17.15.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16 times the desired bit rate.

(3) Transmitting and Receiving Data

- SCIF Initialization (Asynchronous Mode)

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the operating mode or the communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing TE and RE to 0, however, does not initialize the serial status register (SCFSR), transmit FIFO data register (SCFTDR), or receive FIFO data register (SCFRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the SCFSR is set. The TE bit can be cleared to 0 during transmission, but the transmit data goes to the Mark state after the bit is cleared to 0. Set the TFRST bit in SCFCR to 1 and reset SCFTDR before TE is set again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIF operation becomes unreliable if the clock is stopped.

Figure 17.3 shows a sample flowchart for initializing the SCIF.

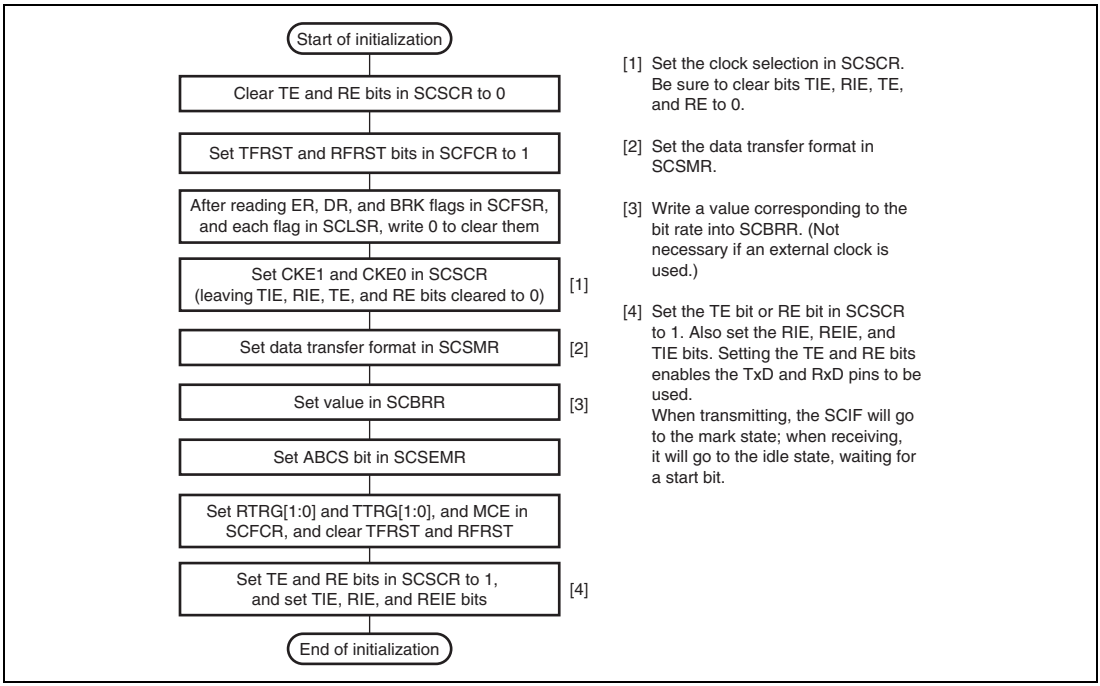


Figure 17.3 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Asynchronous Mode)

Figure 17.4 shows a sample flowchart for serial transmission. Use the following procedure for serial data transmission after enabling the SCIF for transmission.

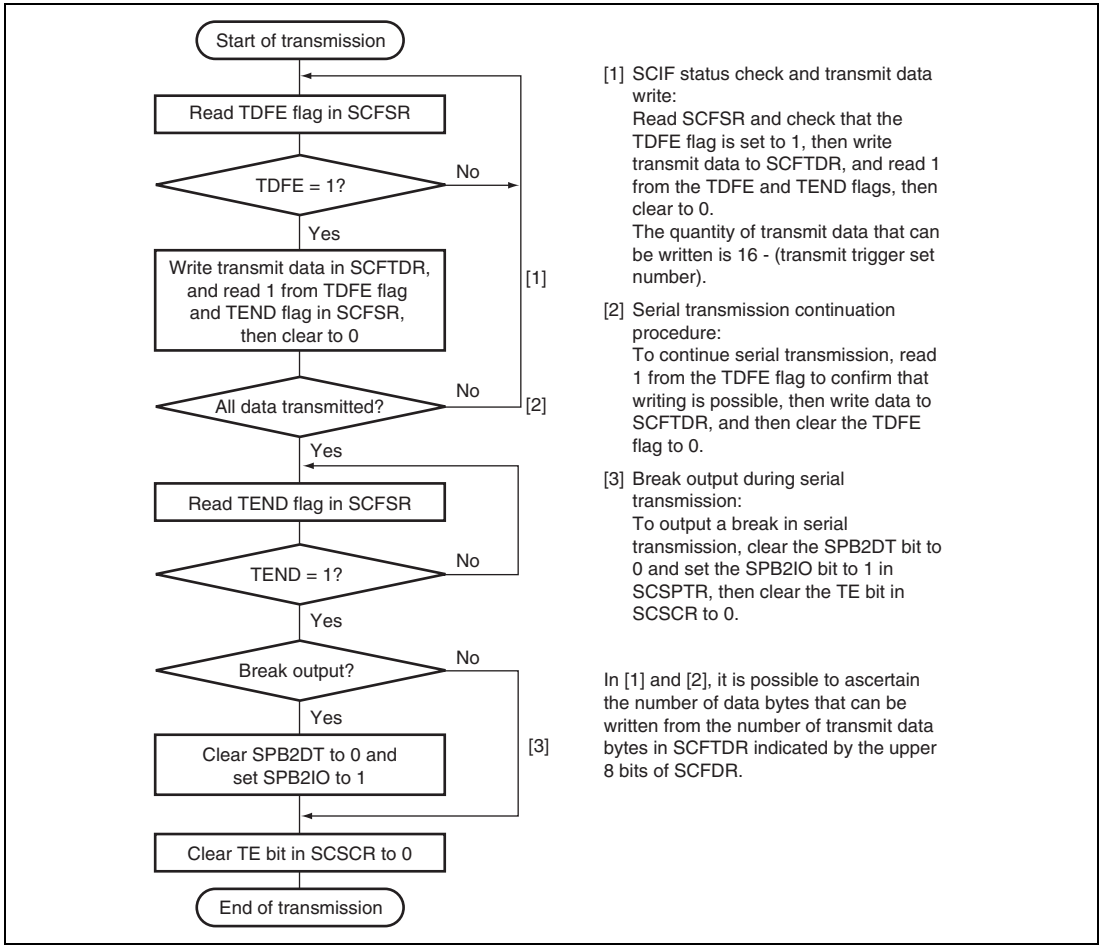


Figure 17.4 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   The serial transmit data is sent from the TXD pin in the following order.

   A. Start bit: One-bit 0 is output.
   B. Transmit data: 8-bit or 7-bit data is output in LSB-first order.
   C. Parity bit: One parity bit (even or odd parity) is output. (A format in which a parity bit is not output can also be selected.)
   D. Stop bit(s): One or two 1 bits (stop bits) are output.
   E. Mark state: 1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCIF checks the SCFTDR transmit data at the timing for sending the stop bit. If data is present, the data is transferred from SCFTDR to SCTSR, the stop bit is sent, and then serial transmission of the next frame is started.

Figure 17.5 shows an example of the operation for transmission.

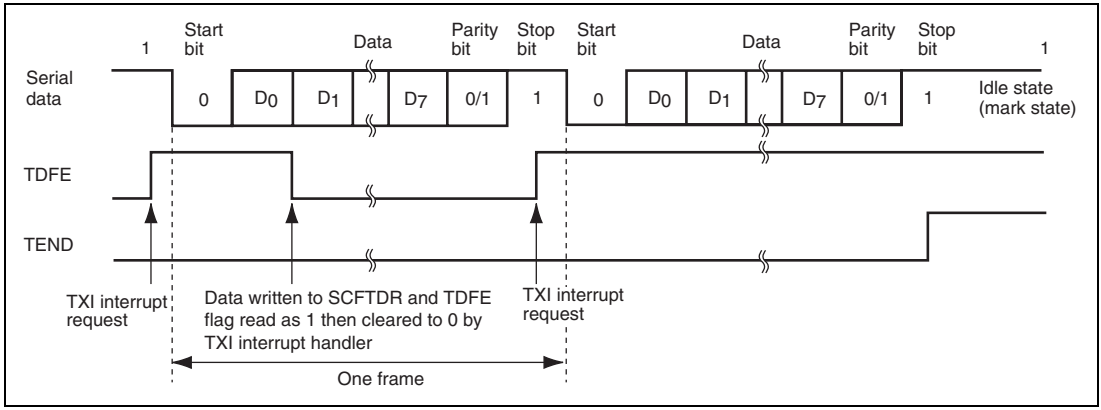
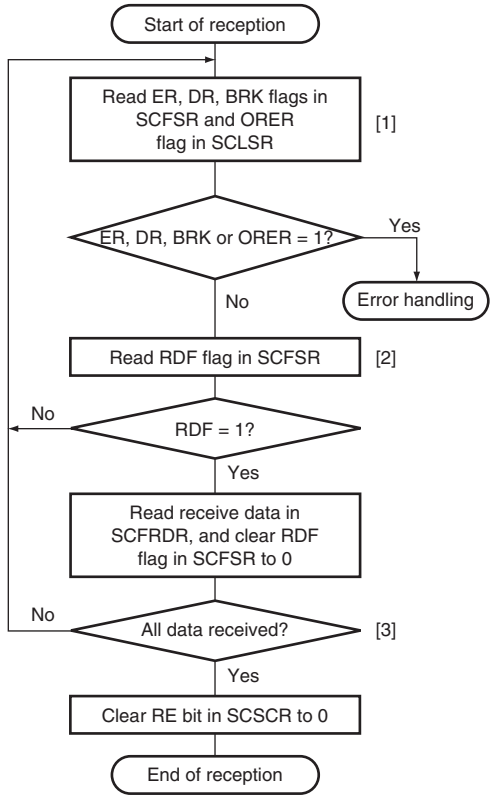


Figure 17.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

- Receiving Serial Data (Asynchronous Mode)

Figures 17.6 and 17.7 show sample flowcharts for serial reception. Use the following procedure for serial data reception after enabling the SCIF for reception.



[1] Receive error handling and break detection:

Read the DR, ER, and BRK flags in SCFSR, and the ORER flag in SCLSR, to identify any error, perform the appropriate error handling, then clear the DR, ER, BRK, and ORER flags to 0. In the case of a framing error, a break can also be detected by reading the value of the RxD pin.

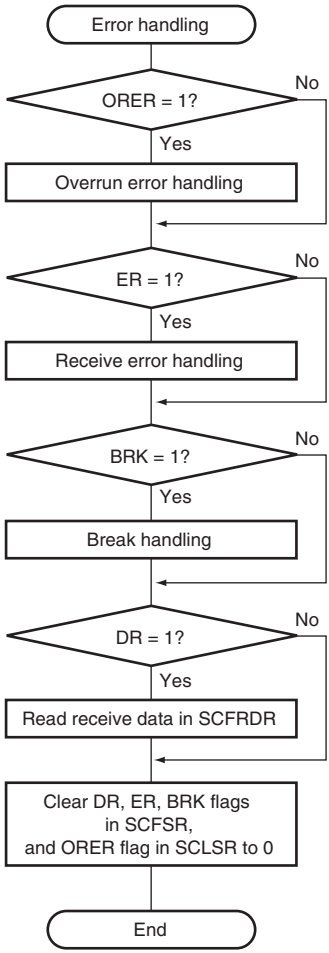
[2] SCIF status check and receive data read:

Read SCFSR and check that RDF flag = 1, then read the receive data in SCFRDR, read 1 from the RDF flag, and then clear the RDF flag to 0. The transition of the RDF flag from 0 to 1 can be identified by an RXI interrupt.

[3] Serial reception continuation procedure:

To continue serial reception, read at least the receive trigger set number of receive data bytes from SCFRDR, read 1 from the RDF flag, then clear the RDF flag to 0. The number of receive data bytes in SCFRDR can be ascertained by reading from SCRFDR.

Figure 17.6 Sample Flowchart for Receiving Serial Data



- Whether a framing error or parity error has occurred in the receive data that is to be read from the receive FIFO data register (SCFRDR) can be ascertained from the FER and PER bits in the serial status register (SCFSR).
- When a break signal is received, receive data is not transferred to SCFRDR while the BRK flag is set. However, note that the last data in SCFRDR is H'00, and the break data in which a framing error occurred is stored.

Figure 17.7 Sample Flowchart for Receiving Serial Data (cont)

In serial reception, the SCIF operates as described below.

1. The SCIF monitors the transmission line, and if a 0 start bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in SCRSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.
   After receiving these bits, the SCIF carries out the following checks.
   A. Stop bit check: The SCIF checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   B. The SCIF checks whether receive data can be transferred from the receive shift register (SCRSR) to SCFRDR.
   C. Overrun check: The SCIF checks that the ORER flag is 0, indicating that the overrun error has not occurred.
   D. Break check: The SCIF checks that the BRK flag is 0, indicating that the break state is not set.

   If all the above checks are passed, the receive data is stored in SCFRDR.

   Note: When a parity error or a framing error occurs, reception is not suspended.
4. If the RIE bit in SCSCR is set to 1 when the RDF or DR flag changes to 1, a receive-FIFO-data-full interrupt (RXI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the ER flag changes to 1, a receive-error interrupt (ERI) request is generated. If the RIE bit or the REIE bit in SCSCR is set to 1 when the BRK or ORER flag changes to 1, a break reception interrupt (BRI) request is generated.

Figure 17.8 shows an example of the operation for reception.

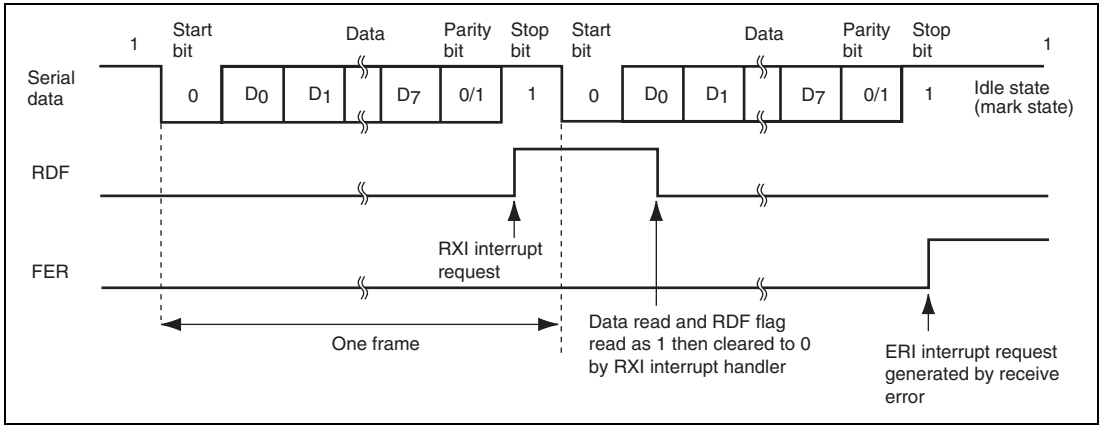


Figure 17.8 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

17.4.3 Operation in Clocked Synchronous Mode

In clocked synchronous mode, the SCIF transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCIF transmitter and receiver are independent, so full-duplex communication is possible while sharing the same clock. The transmitter and receiver are also 16-byte FIFO buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 17.9 shows the general format in clocked synchronous serial communication.

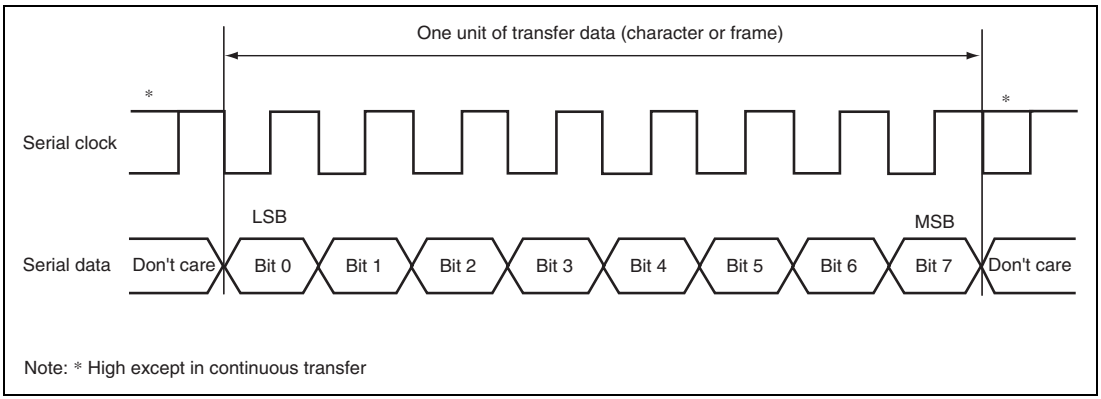


Figure 17.9 Data Format in Clocked Synchronous Communication

In clocked synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

In clocked synchronous mode, the SCIF receives data by synchronizing with the rising edge of the serial clock.

(1) Transmit/Receive Formats

The data length is fixed at eight bits. No parity bit can be added.

(2) Clock

An internal clock generated by the on-chip baud rate generator by the setting of the C/A bit in SCSMR and CKE[1:0] in SCSCR, or an external clock input from the SCK pin can be selected as the SCIF transmit/receive clock.

When the SCIF operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCIF is not transmitting or receiving, the clock signal remains in the high state. When only receiving, the clock signal outputs while the RE bit of SCSCR is 1 and the number of data in receive FIFO is more than the receive FIFO data trigger number.

(3) Transmitting and Receiving Data

- SCIF Initialization (Clocked Synchronous Mode)

Before transmitting, receiving, or changing the mode or communication format, the software must clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF. Clearing TE to 0 initializes the transmit shift register (SCTSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and receive data register (SCRDR), which retain their previous contents.

Figure 17.10 shows a sample flowchart for initializing the SCIF.

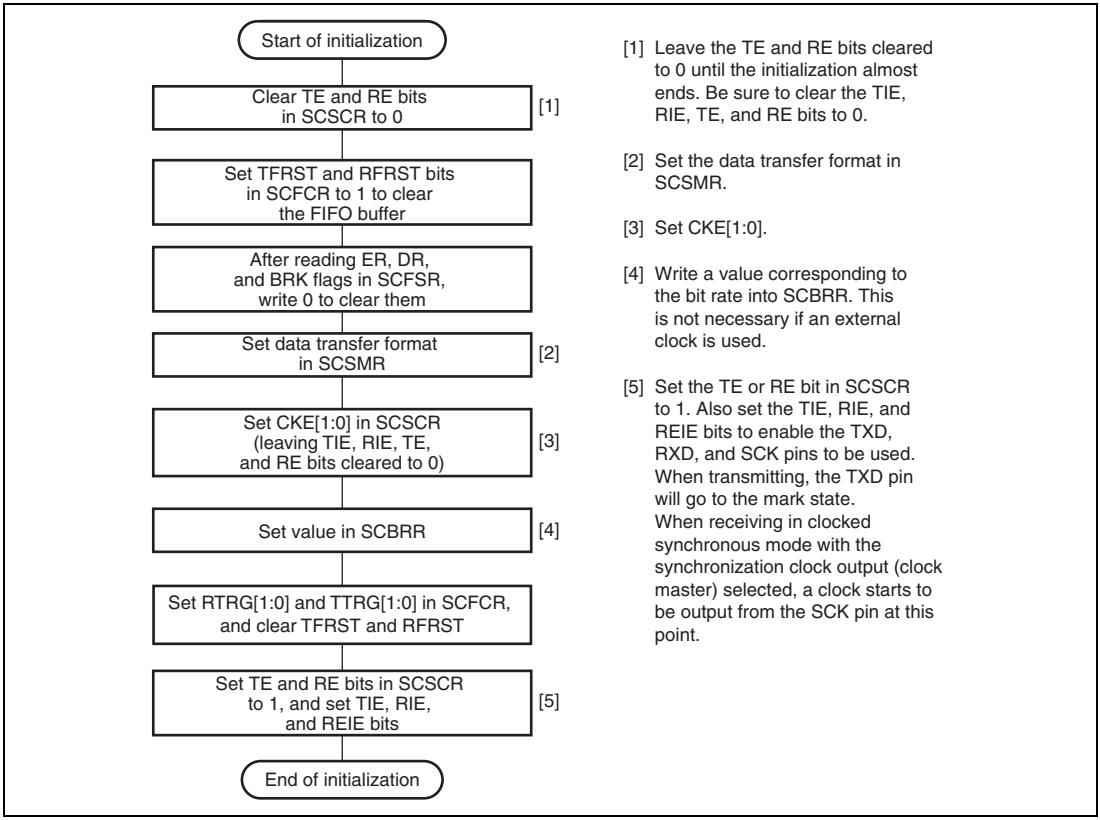


Figure 17.10 Sample Flowchart for SCIF Initialization

- Transmitting Serial Data (Clocked Synchronous Mode)

Figure 17.11 shows a sample flowchart for transmitting serial data. Use the following procedure for serial data transmission after enabling the SCIF for transmission.

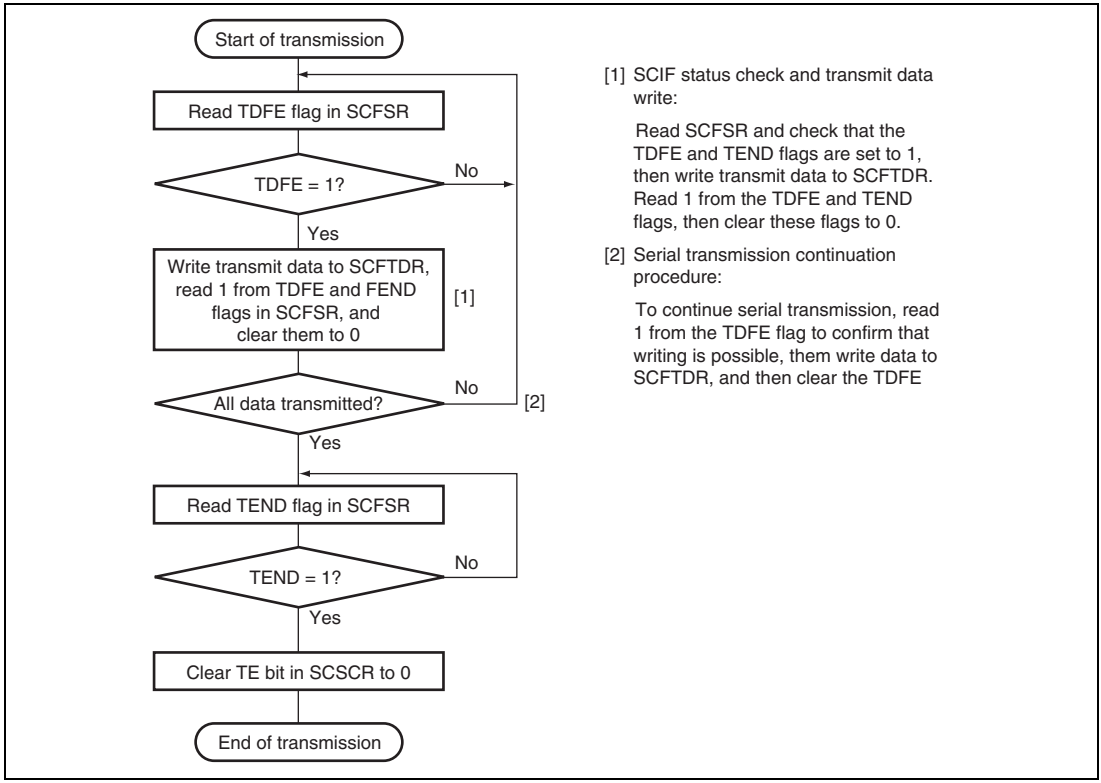


Figure 17.11 Sample Flowchart for Transmitting Serial Data

In serial transmission, the SCIF operates as described below.

1. When data is written into the transmit FIFO data register (SCFTDR), the SCIF transfers the data from SCFTDR to the transmit shift register (SCTSR) and starts transmitting. Confirm that the TDFE flag in the serial status register (SCFSR) is set to 1 before writing transmit data to SCFTDR. The number of data bytes that can be written is (16 – transmit trigger setting).
2. When data is transferred from SCFTDR to SCTSR and transmission is started, consecutive transmit operations are performed until there is no transmit data left in SCFTDR. When the number of transmit data bytes in SCFTDR falls below the transmit trigger number set in the FIFO control register (SCFCR), the TDFE flag is set. If the TIE bit in the serial control register (SCSR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.

   If clock output mode is selected, the SCIF outputs eight synchronous clock pulses. If an external clock source is selected, the SCIF outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7).
3. The SCIF checks the SCFTDR transmit data at the timing for sending the MSB (bit 7). If data is present, the data is transferred from SCFTDR to SCTSR, and then serial transmission of the next frame is started. If there is no data, the TXD pin holds the state after the TEND flag in SCFSR is set to 1 and the MSB (bit 7) is sent.
4. After the end of serial transmission, the SCK pin is held in the high state.

Figure 17.12 shows an example of SCIF transmit operation.

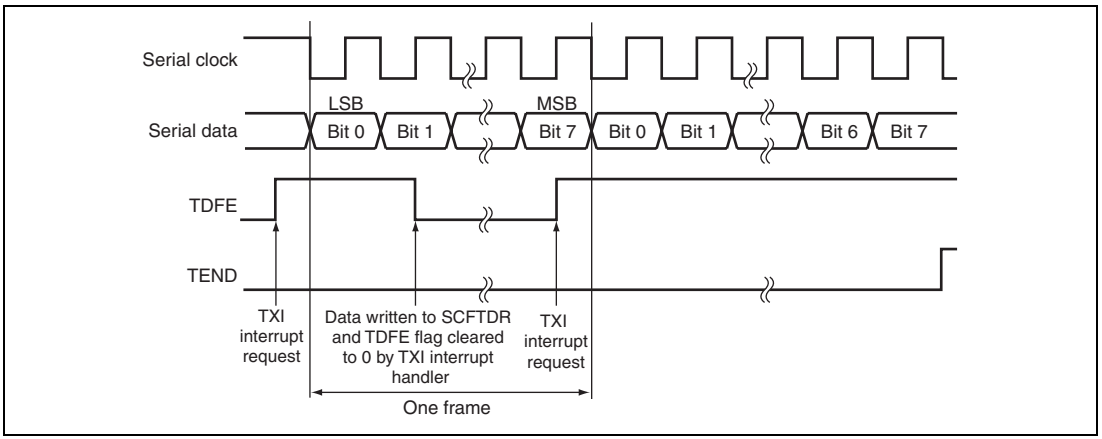


Figure 17.12 Example of SCIF Transmit Operation

- Receiving Serial Data (Clocked Synchronous Mode)

Figures 17.13 and 17.14 show sample flowcharts for receiving serial data. When switching from asynchronous mode to clocked synchronous mode without SCIF initialization, make sure that ORER, PER, and FER are cleared to 0.

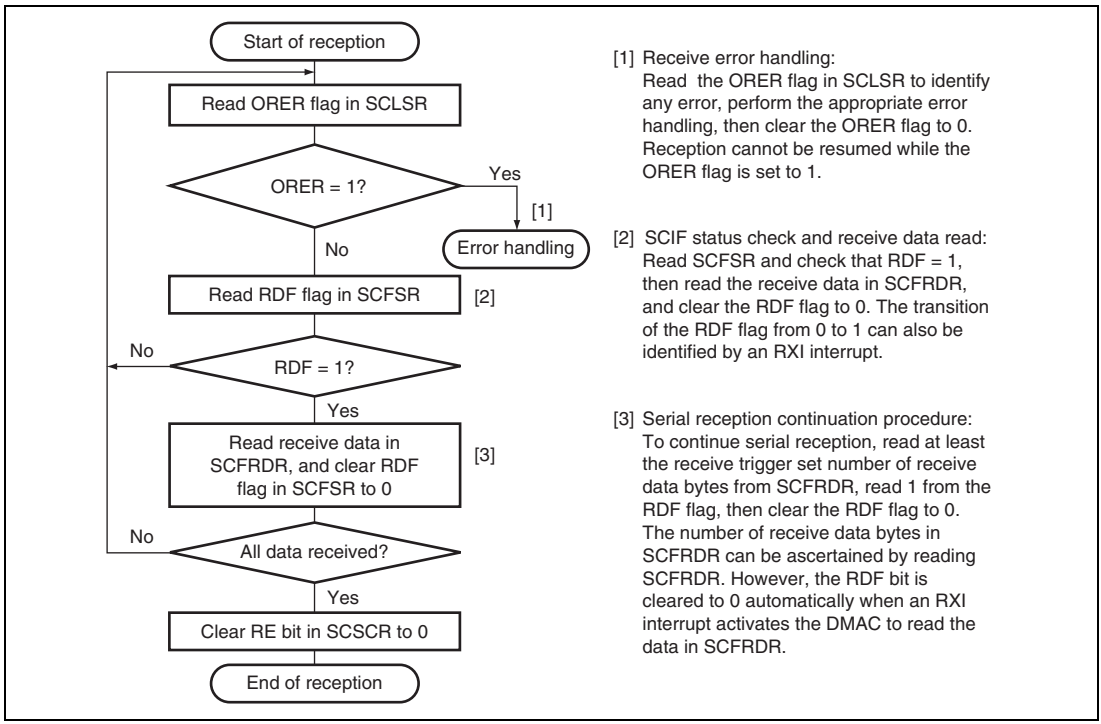


Figure 17.13 Sample Flowchart for Receiving Serial Data (1)

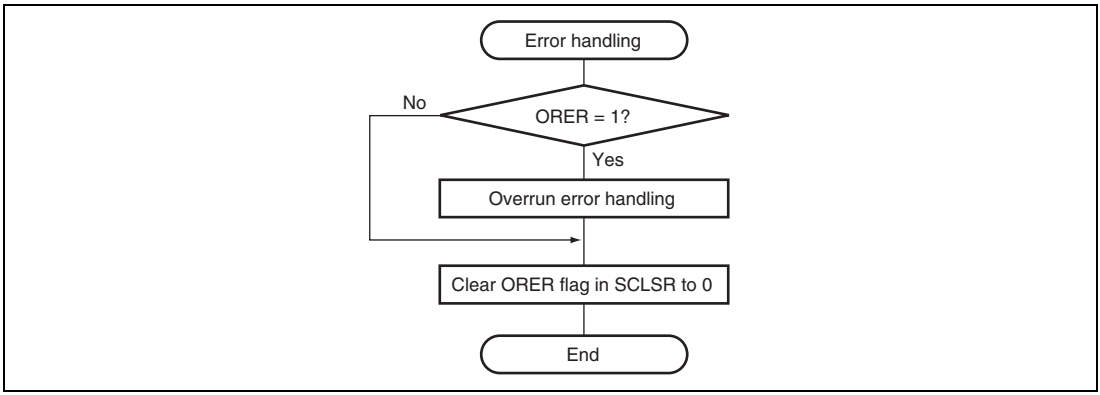


Figure 17.14 Sample Flowchart for Receiving Serial Data (2)

In serial reception, the SCIF operates as described below.

1. The SCIF synchronizes with serial clock input or output and starts the reception.
2. Receive data is shifted into SCRSR in order from the LSB to the MSB. After receiving the data, the SCIF checks the receive data can be loaded from SCRSR into SCFRDR or not. If this check is passed, the RDF flag is set to 1 and the SCIF stores the received data in SCFRDR. If the check is not passed (overrun error is detected), further reception is prevented.
3. After setting RDF to 1, if the receive FIFO data full interrupt enable bit (RIE) is set to 1 in SCSCR, the SCIF requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) or the receive error interrupt enable bit (REIE) in SCSCR is also set to 1, the SCIF requests a break interrupt (BRI).

Figure 17.15 shows an example of SCIF receive operation.

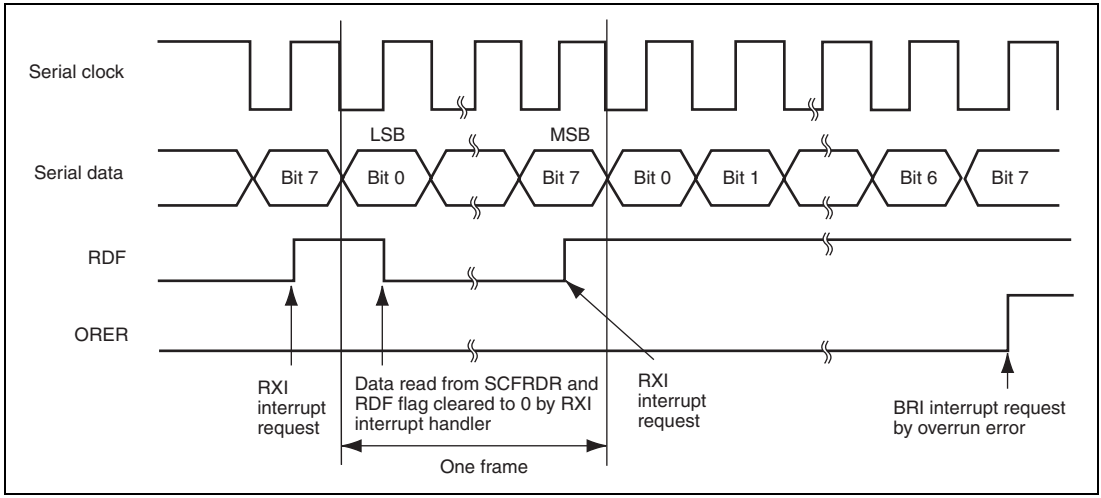


Figure 17.15 Example of SCIF Receive Operation

- Transmitting and Receiving Serial Data Simultaneously (Clocked Synchronous Mode)

Figure 17.16 shows a sample flowchart for transmitting and receiving serial data simultaneously. Use the following procedure for the simultaneous transmission/reception of serial data, after enabling the SCIF for transmission/reception.

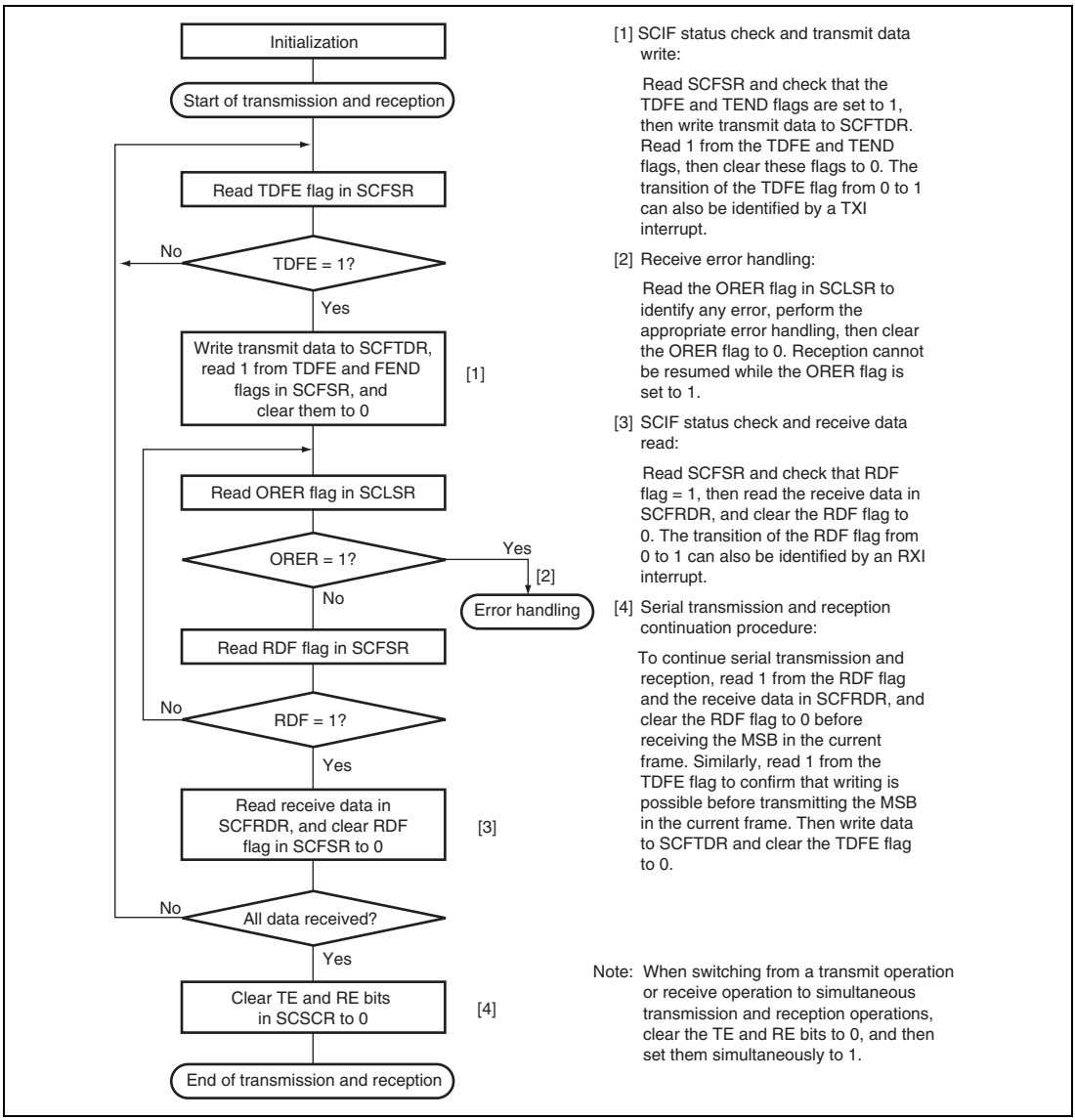


Figure 17.16 Sample Flowchart for Transmitting/Receiving Serial Data

17.5 SCIF Interrupts

The SCIF has four interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive FIFO data full (RXI), and break (BRI).

Table 17.17 shows the interrupt sources and their order of priority. The interrupt sources are enabled or disabled by means of the TIE, RIE, and REIE bits in SCSCR. A separate interrupt request is sent to the interrupt controller for each of these interrupt sources.

When a TXI request is enabled by the TIE bit and the TDFE flag in the serial status register (SCFSR) is set to 1, a TXI interrupt request is generated. The DMAC or DTC can be activated and data transfer performed by this TXI interrupt request. At DMAC activation, an interrupt request is not sent to the CPU.

When an RXI request is enabled by the RIE bit and the RDFE flag or the DR flag in SCFSR is set to 1, an RXI interrupt request is generated. The DMAC or DTC can be activated and data transfer performed by this RXI interrupt request. At DMAC activation, an interrupt request is not sent to the CPU. The RXI interrupt request caused by the DR flag is generated only in asynchronous mode.

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests only an ERI interrupt without requesting an RXI interrupt.

The TXI interrupt indicates that transmit data can be written, and the RXI interrupt indicates that there is receive data in SCFRDR.

Table 17.17 SCIF Interrupt Sources

Interrupt Source	Description	DMAC or DTC Activation	Priority on Reset Release
BRI	Interrupt initiated by break (BRK) or overrun error (ORER)	Not possible	High
ERI	Interrupt initiated by receive error (ER)	Not possible	
RXI	Interrupt initiated by receive FIFO data full (RDF) or data ready (DR)	Possible	
TXI	Interrupt initiated by transmit FIFO data empty (TDFE)	Possible	Low

17.6 Usage Notes

Note the following when using the SCIF.

17.6.1 SCFTDR Writing and TDFE Flag

The TDFE flag in the serial status register (SCFSR) is set when the number of transmit data bytes written in the transmit FIFO data register (SCFTDR) has fallen below the transmit trigger number set by bits TTRG[1:0] in the FIFO control register (SCFCR). After the TDFE flag is set, transmit data up to the number of empty bytes in SCFTDR can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in SCFTDR is equal to or less than the transmit trigger number, the TDFE flag will be set to 1 again after being read as 1 and cleared to 0. TDFE flag clearing should therefore be carried out when SCFTDR contains more than the transmit trigger number of transmit data bytes.

The number of transmit data bytes in SCFTDR can be found from the upper 8 bits of the FIFO data count register (SCFDR).

17.6.2 SCFRDR Reading and RDF Flag

The RDF flag in the serial status register (SCFSR) is set when the number of receive data bytes in the receive FIFO data register (SCFRDR) has become equal to or greater than the receive trigger number set by bits RTRG[1:0] in the FIFO control register (SCFCR). After RDF flag is set, receive data equivalent to the trigger number can be read from SCFRDR, allowing efficient continuous reception.

However, if the number of data bytes in SCFRDR exceeds the trigger number, the RDF flag will be set to 1 again if it is cleared to 0. The RDF flag should therefore be cleared to 0 after being read as 1 after reading the number of the received data in the receive FIFO data register (SCFRDR) which is less than the trigger number.

The number of receive data bytes in SCFRDR can be found from the lower 8 bits of the FIFO data count register (SCFDR).

17.6.3 Restriction on DMAC and DTC Usage

When the DMAC or DTC writes data to SCFTDR due to a TXI interrupt request, the state of the TEND flag becomes undefined. Therefore, the TEND flag should not be used as the transfer end flag in such a case.

17.6.4 Break Detection and Processing

Break signals can be detected by reading the RXD pin directly when a framing error (FER) is detected. In the break state the input from the RXD pin consists of all 0s, so the FER flag is set and the parity error flag (PER) may also be set.

Note that, although transfer of receive data to SCFRDR is halted in the break state, the SCIF receiver continues to operate.

17.6.5 Sending a Break Signal

The I/O condition and level of the TXD pin are determined by the SPB2IO and SPB2DT bits in the serial port register (SCSPTR). This feature can be used to send a break signal.

Until TE bit is set to 1 (enabling transmission) after initializing, the TXD pin does not work. During the period, mark status is performed by the SPB2DT bit. Therefore, the SPB2IO and SPB2DT bits should be set to 1 (high level output).

To send a break signal during serial transmission, clear the SPB2DT bit to 0 (designating low level), then clear the TE bit to 0 (halting transmission). When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, and 0 is output from the TXD pin.

17.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode)

The SCIF operates on a base clock with a frequency of 16 times the transfer rate.* In reception, the SCIF synchronizes internally with the fall of the start bit, which it samples on the base clock. Receive data is latched at the rising edge of the eighth base clock pulse. The timing is shown in figure 17.17.

Note: * This is an example when ABCS = 0 in SCSEMR. When ABCS = 1, a frequency of 8 times the bit rate becomes the basic clock, and receive data is sampled at the fourth rising edge of the basic clock.

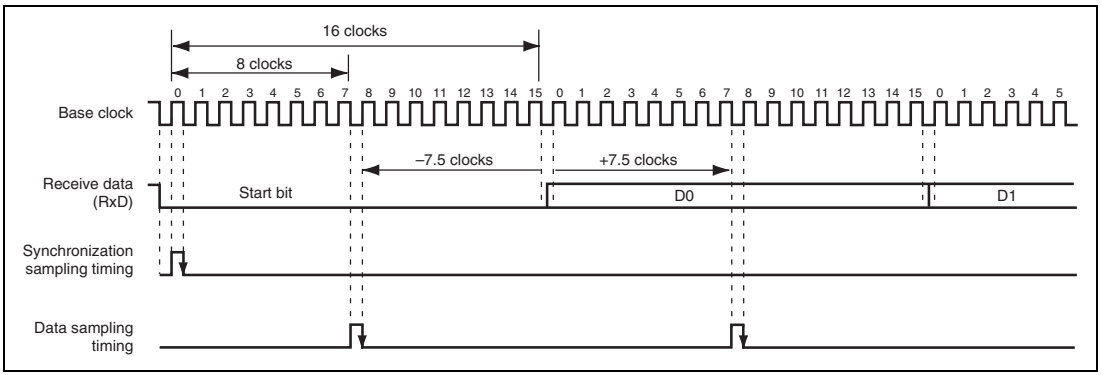


Figure 17.17 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$

$$= 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

17.6.7 FER Flag and PER Flag of Serial Status Register (SCFSR)

The FER flag and PER flag in the serial status register (SCFSR) are status flag that apply to next entry to be read from the receive FIFO data register (SCFRDR). After the CPU or DTC/DMAC reads the receive FIFO data register, the flags of framing errors and parity errors will disappear.

To check the received data for the states of framing errors and parity errors, only read the receive FIFO register after reading the serial status register.

Section 18 Renesas Serial Peripheral Interface (RSPI)

This LSI includes a channel of Renesas Serial Peripheral Interface (RSPI).

The RSPI is capable of full-duplex synchronous, high-speed serial communications with multiple processors and peripheral devices.

18.1 Features

The RSPI of this LSI has the following features:

1. RSPI Transfer Function
- Uses MOSI (Master Out Slave In), MISO (Maser In Slave Out), SSL (Slave Select), and RSPCK (RSPI Clock) signals to provide SPI mode (four-wire) and clock synchronous mode (three-wire) serial communications.
- Capable of master-slave mode serial communication.
- Capable of mode fault error detection.
- Capable of overrun error detection.
- Modifiable serial transfer clock polarity.
- Modifiable serial transfer clock phase.

2. Data Format
- Switchable MSB first/LSB first.
- Transfer bit length changeable to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, and 32 bits.
- Transmission/receive buffers of 128 bits
- Up to 4 frames (up to 32 bits per frame) can be transferred at a time in transmission or reception.

3. Bit Rate
- In master mode:
  An internal baud rate generator generates RSPCK by dividing Pϕ by up to 4906.
- In slave mode:
  The serial clock signal is generated with division by up to 8.
  An external input clock is used as the serial clock.

4. Buffer Configuration

- Transmission/receive buffers are provided in a double-buffer configuration.

5. SSL Control Function

- Provided with four SSL signals (SSL0 to SSL3).
- In single-master mode, SSL0 to SSL3 signals are for output.
- In multi-master mode, SSL0 signal is for input, and SSL1 to SSL3 signals are for either output or Hi-Z.
- In slave mode, SSL0 signal is for input, and SSL1 to SSL3 signals are for Hi-Z.
- A delay from SSL output assertion to RSPCK operation (RSPCK delay) can be set.
  Settable range: 1 to 8 RSPCK cycles
  Unit: 1 RSPCK cycle
- A delay from RSPCK stop to SSL output negation (SSL negation delay) can be set.
  Settable range: 1 to 8 RSPCK cycles
  Unit: 1 RSPCK cycle
- Wait for next-access SSL output assertion (next-access delay) can be set.
  Settable range: 1 to 8 RSPCK cycles
  Unit: 1 RSPCK cycle
- Switchable SSL polarity.

6. Master Mode Transfer Control Method

- A transfer comprised of a maximum of four commands can be executed in sequential loops.
- Each command can include:
  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 started upon writing to the transmit buffer by the DMAC.
- A transfer can be started upon writing to the transmit buffer by the DTC.
- A transfer can be started upon clearing the SPTEF bit by the CPU.
- MOSI signal values can be set during SSL negation.

7. Interrupt Sources

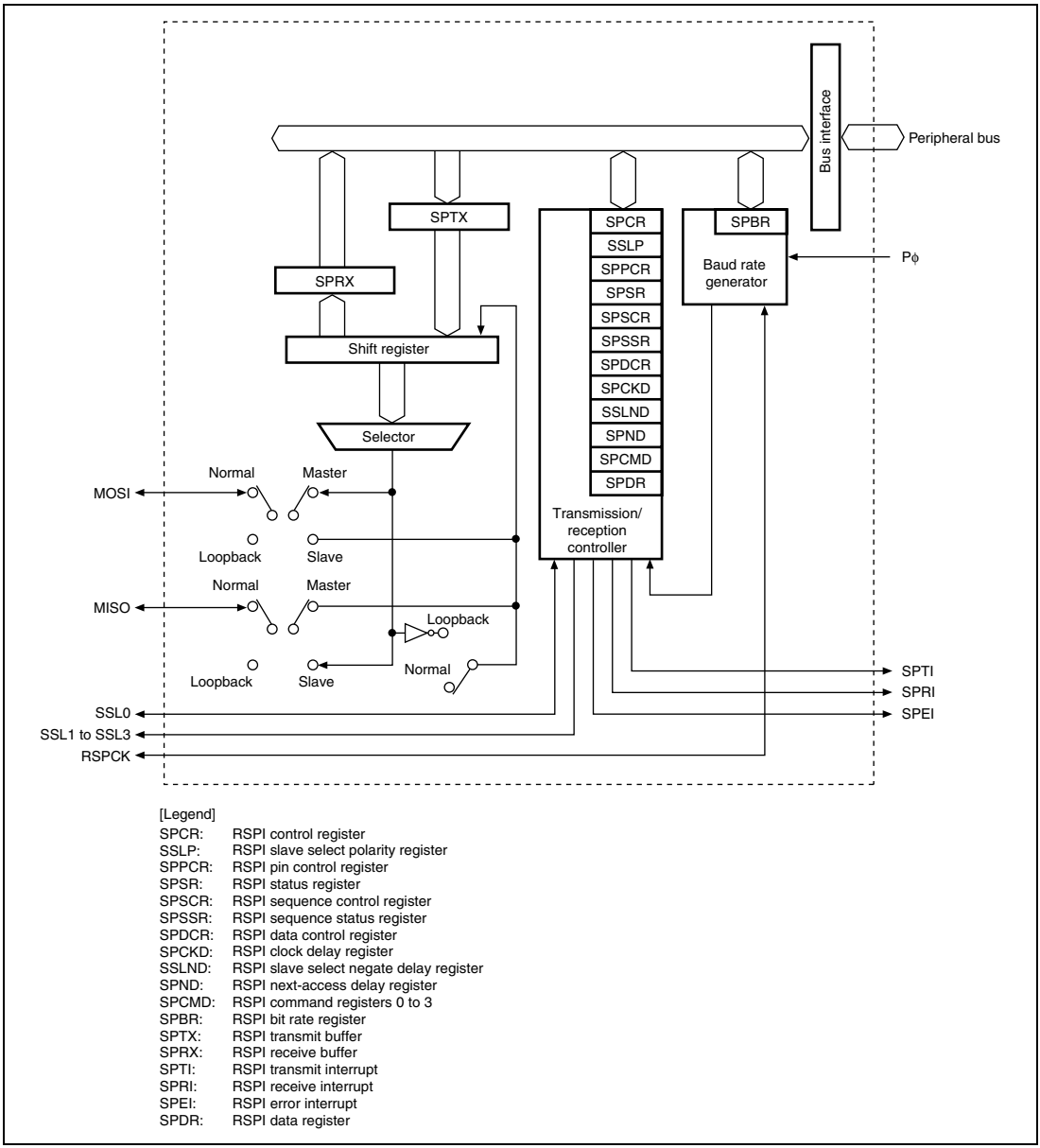
- Maskable interrupt sources are provided.
  - RSPI receive interrupt (receive buffer full)
  - RSPI transmit interrupt (transmit buffer empty)
  - RSPI error interrupt (mode fault and overrun)

8. Other Features

- Loopback mode is provided.
- The CMOS/open drain output switchover function is provided.
- The RSPI disable (initialization) function is provided.

18.1.1 Internal Block Diagram

Figure 18.1 shows an RSPI block diagram.



[Legend]

SPCR:	RSPI control register
SSLP:	RSPI slave select polarity register
SPPCR:	RSPI pin control register
SPSR:	RSPI status register
SPSCR:	RSPI sequence control register
SPSSR:	RSPI sequence status register
SPDCR:	RSPI data control register
SPCKD:	RSPI clock delay register
SSLND:	RSPI slave select negate delay register
SPND:	RSPI next-access delay register
SPCMD:	RSPI command registers 0 to 3
SPBR:	RSPI bit rate register
SPTX:	RSPI transmit buffer
SPRX:	RSPI receive buffer
SPTI:	RSPI transmit interrupt
SPRI:	RSPI receive interrupt
SPEI:	RSPI error interrupt
SPDR:	RSPI data register

Figure 18.1 Block Diagram of RSPI

18.2 Input/Output Pins

The RSPI has the serial pins shown in table 18.1. The RSPI automatically switches input/output directions of the pins. Pin SSL0 is set to output when the RSPI is in single master mode and set to input when the RSPI is in multi master or slave mode. Pins RSPCK, MOSI, and MISO are set to inputs or outputs according to the master/slave setting and input level of SSL0 (see section 18.4.2, Controlling RSPI Pins).

Table 18.1 Pin Configuration

Pin Name	Symbol	I/O	Function
RSPI clock pin	RSPCK	I/O	RSPI clock input/output
Master transmit data pin	MOSI	I/O	RSPI master transmit data
Slave transmit data pin	MISO	I/O	RSPI slave transmit data
Slave select 0 pin	SSL0	I/O	RSPI slave select
Slave select 1 pin	SSL1	Output	RSPI slave select
Slave select 2 pin	SSL2	Output	RSPI slave select
Slave select 3 pin	SSL3	Output	RSPI slave select

Note: Pin names RSPCK, MOSI, MISO, and SSL0 to SSL3 are used in the description for all channels, omitting the channel designation.

18.3 Register Descriptions

The RSPI has the registers shown in table 18.2. 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 18.2 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
RSPI control register	SPCR	R/W	H'00	H'FFFFB000	8, 16
RSPI slave select polarity register	SSLP	R/W	H'00	H'FFFFB001	8
RSPI pin control register	SPPCR	R/W	H'00	H'FFFFB002	8, 16
RSPI status register	SPSR	R/W	H'22	H'FFFFB003	8
RSPI data register	SPDR	R/W	H'00000000	H'FFFFB004	16, 32*
RSPI sequence control register	SPSCR	R/W	H'00	H'FFFFB008	8, 16
RSPI sequence status register	SPSSR	R	H'00	H'FFFFB009	8
RSPI bit rate register	SPBR	R/W	H'FF	H'FFFFB00A	8, 16
RSPI data control register	SPDCR	R/W	H'00	H'FFFFB00B	8
RSPI clock delay register	SPCKD	R/W	H'00	H'FFFFB00C	8, 16
RSPI slave select negation delay register	SSLND	R/W	H'00	H'FFFFB00D	8
RSPI next-access delay register	SPND	R/W	H'00	H'FFFFB00E	8
RSPI command register 0	SPCMD0	R/W	H'070D	H'FFFFB010	16
RSPI command register 1	SPCMD1	R/W	H'070D	H'FFFFB012	16
RSPI command register 2	SPCMD2	R/W	H'070D	H''FFFFB014	16
RSPI command register 3	SPCMD3	R/W	H'070D	H'FFFFB016	16

Notes: * Use the access size set by the SPLW bit.

18.3.1 RSPI Control Register (SPCR)

SPCR sets the operating mode of the RSPI. SPCR can be read from or written to by the CPU. If the MSTR and MODFEN bits are changed while the RSPI function is enabled by setting the SPE bit to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	–	SPMS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	SPRIE	0	R/W	RSPI Receive Interrupt Enable If the RSPI has detected a receive buffer write after completion of a serial transfer and the SPRF bit in the RSPI status register (SPSR) is set to 1, this bit enables or disables the generation of an RSPI receive interrupt request. 0: Disables the generation of RSPI receive interrupt requests. 1: Enables the generation of RSPI receive interrupt requests.
6	SPE	0	R/W	RSPI Function Enable Setting this bit to 1 enables the RSPI function. When the MODF bit in the RSPI status register (SPSR) is 1, the SPE bit cannot be set to 1 (see section 18.4.7, Error Detection). Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function (see section 18.4.8, Initializing RSPI). 0: Disables the RSPI function 1: Enables the RSPI function

Bit	Bit Name	Initial Value	R/W	Description
5	SPTIE	0	R/W	RSPI Transmit Interrupt Enable Enables or disables the generation of RSPI transmit interrupt requests when the RSPI detects transmit buffer empty and sets the SPTEF bit in the RSPI status register (SPSR) to 1. In the RSPI disabled (with the SPE bit 0) status, the SPTEF bit is 1. Therefore, note that setting the SPTIE bit to 1 when the RSPI is in the disabled status generates an RSPI transmit interrupt request. 0: Disables the generation of RSPI transmit interrupt requests. 1: Enables the generation of RSPI transmit interrupt requests.
4	SPEIE	0	R/W	RSPI Error Interrupt Enable Enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the MODF bit in the RSPI status register (SPSR) to 1, or when the RSPI detects and sets the OVRF bit in SPSR to 1 (see section 18.4.7, Error Detection). 0: Disables the generation of RSPI error interrupt requests. 1: Enables the generation of RSPI error interrupt requests.
3	MSTR	0	R/W	RSPI Master/Slave Mode Select Selects master/slave mode of RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCK, MOSI, MISO, and SSL0 to SSL3. 0: Slave mode 1: Master mode
2	MODFEN	0	R/W	Mode Fault Error Detection Enable Enables or disables the detection of mode fault error (see section 18.4.7, Error Detection). In addition, the RSPI determines the input/output directions of the SSL0 pin based on combinations of the MODFEN and MSTR bits (see section 18.4.2, Controlling RSPI Pins). 0: Disables the detection of mode fault error 1: Enables the detection of mode fault error

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SPMS	0	R/W	RSPI Mode Select Selects SPI (4-wire) or clock synchronous (3-wire) mode. In clock synchronous mode, the SSL pin is not used and the RSPCK, MOSI, and MISO pins are used for communication. To enable clock synchronous mode, set the CPHA bit in the RSPI command register (SPCMD) to 1. If CPHA is set to 0, operation cannot be guaranteed. 0: SPI mode (4-wire) 1: Clock synchronous mode

18.3.2 RSPI Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL0 to SSL7 signals of the RSPI. SSLP can always be read from or written to by the CPU. If the contents of SSLP are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	SSL3P	SSL2P	SSL1P	SSL0P
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SSL3P	0	R/W	SSL Signal Polarity Setting These bits set the polarity of the SSL signals. SSLiP (where i is 3 to 0) indicates the active polarity of the SSLi signal. 0: SSLi signal set to active-0 1: SSLi signal set to active-1
2	SSL2P	0	R/W	
1	SSL1P	0	R/W	
0	SSL0P	0	R/W	

18.3.3 RSPI Pin Control Register (SPPCR)

SPPCR sets the modes of the RSPI pins. SPPCR can be read from or written to by the CPU. If the contents of this register are changed by the CPU while the RSPI function is enabled by setting the SPE bit in the RSPI control register (SPCR) to 1, operation cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	MOIFE	MOIFV	–	SPOM	–	SPLP
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5	MOIFE	0	R/W	MOSI Idle Value Fixing Enable 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 MOIFE is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period. When MOIFE is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSI bit. 0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit
4	MOIFV	0	R/W	MOSI Idle Fixed Value If the MOIFE bit is 1 in master mode, the RSPI, according to MOIFV bit settings, determines the MOSI signal value during the SSL negation period (including the SSL retention period during a burst transfer). 0: MOSI Idle fixed value equals 0 1: MOSI Idle fixed value equals 1
3	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
2	SPOM	0	R/W	RSPI Output Pin Mode Sets the RSPI output pins to CMOS output/open drain output. 0: CMOS output 1: Open-drain output
1	—	0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
0	SPLP	0	R/W	RSPI Loopback When the SPLP bit is set to 1, the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects (reverses) the input path and the output path for the shift register (loopback mode). 0: Normal mode 1: Loopback mode

18.3.4 RSPI Status Register (SPSR)

SPSR indicates the operating status of the RSPI. SPSR can be read by the CPU. Writing 1 to the SPRF, SPTEF, MODF, and OVRF bits cannot be performed by the CPU. These bits can be cleared to 0 after they are read as 1.

Bit:	7	6	5	4	3	2	1	0
	SPRF	–	SPTEF	–	–	MODF	MIDLE	OVRF
Initial value:	0	0	1	0	0	0	0	0
R/W:	R/(W)*	R	R/(W)*	R	R	R/(W)*	R	R/(W)*

Note: * Only 0 can be written to this bit after reading it as 1 to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	SPRF	0	R/(W)*	RSPI Receive Buffer Full Flag Indicates the status of the receive buffer for the RSPI data register (SPDR). Upon completion of a serial transfer with the SPRF bit 0, the RSPI transfers the receive data from the shift register to SPDR, and sets this bit to 1. This also means that the last bit of transmit data has been sent because the RSPI performs full-duplex synchronous serial communication. If a serial transfer ends while the SPRF bit is 1, the RSPI does not transfer the received data from the shift register to SPDR. When the OVRF bit in SPSR is 1, the SPRF bit cannot be changed from 0 to 1 (see section 18.4.7, Error Detection). 0: No valid data in SPDR [Clearing conditions] • When 0 is written in SPRF after reading SPRF = 1. • When the DMAC is activated with an RXI interrupt and the DMAC reads data from SPDR as many as the number of states specified in SPFC. • When the DTC is activated with an RXI interrupt and the DTC reads data from SPDR as many as the number of states specified in SPFC (except when the transfer counter value of the DTC becomes H'0000 and the DISEL bit is 1). • Power-on reset 1: Valid data found in SPDR [Setting condition] • When serial reception of data as many as the number of states specified in SPFC is normally completed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SPTEF	1	R/(W)*	RSPI Transmit Buffer Empty Flag Indicates the status of the transmit buffer for the RSPI data register (SPDR). When the SPTEF bit is cleared and the shift register is empty, the data is copied from the transmit buffer to the shift register. The CPU, DMAC and DTC can write to SPDR only when the SPTEF bit is 1. If the CPU, the DMAC or the DTC writes to the transmit buffer of SPDR when the SPTEF bit is 0, the data in the transmit buffer is not updated. 0: Data found in the transmit buffer [Clearing conditions] • When 0 is written in SPTEF after reading SPTEF = 1. • When the DMAC is activated with a TXI interrupt and the DMAC writes data to SPDR as many as the number of states specified in SPFC. • When the DTC is activated with a TXI interrupt and the DTC writes data to SPDR as many as the number of states specified in SPFC (except when the transfer counter value of the DTC becomes H'0000 and the DISEL bit is 1). 1: No data in the transmit buffer [Setting conditions] • Power-on reset • When serial reception of data as many as the number of states specified in SPFC is normally completed.
4, 3	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	MODF	0	R/(W)*	Mode Fault Error Flag Indicates the occurrence of a mode fault error. The active level of the SSL0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP). 0: No mode fault error occurs [Clearing conditions] • Power-on reset • When 0 is written in MODF after reading MODF = 1. 1: A mode fault error occurs [Setting conditions] • When the input of SSL0 is set to the active level in multi-master mode. • When the SSL0 pin is negated before the RSPCK cycle necessary for data transfer ends in slave mode
1	MIDLE	1	R	RSPI Idle Flag Indicates the status of RSPI transfer. 1: RSPI is in the idle state. [Setting conditions] In master mode: • The SPE bit in SPCR is 0 (RSPI initialization) • The SPTEF bit in SPSR is 1, the SPSSR bits in SPCP are 00, and the RSPI internal sequencer becomes idle. In slave mode: • The SPE bit in SPCR is 0. 0: RSPI transfers the data. [Clearing condition] When the setting condition is not satisfied.

Bit	Bit Name	Initial Value	R/W	Description
0	OVRF	0	R/(W)*	Overrun Error Flag Indicates the occurrence of an overrun error. 0: No overrun error occurs [Clearing conditions] • Power-on reset • When 0 is written in OVRF after reading OVRF = 1. 1: An overrun error occurs [Setting condition] • When serial transfer is ended while the SPRF bit is set to 1.

Note: * Only 0 can be written to this bit after reading it as 1 to clear the flag.

18.3.5 RSPI Data Register (SPDR)

SPDR is a buffer that stores RSPI transmit/receive data. The transmit buffer (SPTX) and receive buffer (SPRX) are allocated for SPDR and these buffers are independent of each other.

Data should be read from or written to SPDR in word or longword units according to the setting of the RSPI longword/word access setting bit (SPLW) in the RSPI data control register (SPDCR). When the SPLW bit is 0, SPDR is a 64-bit buffer consisting of 4 frames, each of which includes up to 16 bits. When the SPLW bit is 1, SPDR is a 128-bit buffer consisting of 4 frames, each of which includes up to 32 bits.

This register acts as the interface with the FIFO buffer. To read four frames of data, reading SPDR four times will lead to the data being read out in the order of reception. To transmit four frames of data, write to SPDR four times.

The frame length that SPDR uses is determined by the frame count setting bits (SPFC1 and SPFC0) in the RSPI data control register (SPDCR). The bit length to be used is determined by the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD).

If the CPU, DTC, or DMAC requests writing to SPDR when the SPTEF bit in the RSPI status register (SPSR) is 1, the RSPI writes data to the transmit buffer of SPDR. If the SPTEF bit is 0, the RSPI does not update the transmit buffer of SPDR.

When the CPU, DTC, or DMAC requests reading from SPDR, data is read from the receive buffer if the RSPI receive/transmit data select bit (SPRDTD) in the RSPI pin control register (SPPCR) is 0, or data is read from the transmit buffer if the SPRDTD bit is 1.

When reading data from the transmit buffer, the most recently written value is read. If the SPTEF bit in the RSPI status register (SPSR) is 0, no data is read from the transmit buffer.

In the normal operating method, the CPU, DTC, and DMAC read the receive buffer when the SPRF bit in SPSR is 1 (a condition in which unread data is stored in the receive buffer). When the SPRF or OVRF bit in SPSR is 1, the RSPI does not update the receive buffer of SPDR at the end of a serial transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.3.6 RSPI Sequence Control Register (SPSCR)

SPSCR sets the sequence control method when the RSPI operates in master mode. SPSCR can be read from or written to by the CPU. If the contents of SPSCR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function enabled, the subsequent operation cannot be guaranteed.

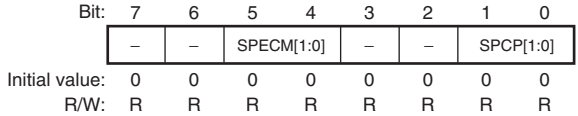
Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	SPSLN[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
1, 0	SPSLN[1:0]	00	R/W	RSPI Sequence Length Setting These bits set a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes RSPI command registers 0 to 3 (SPCMD0 to SPCMD3) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN1 and SPSLN0 bits. When the RSPI is in slave mode, SPCMD0 is always referenced. The relationship among the setting in these bits, sequence length, and referenced SPCMD register number is shown below.

SPSLN [1:0]	Sequence Length	Referenced SPCMD #
00	1	0 → 0 → …
01	2	0 → 1 → 0 → …
10	3	0 → 1 → 2 → 0 → …
11	4	0 → 1 → 2 → 3 → 0 → …

18.3.7 RSPI Sequence Status Register (SPSSR)

SPSSR indicates the sequence control status when the RSPI operates in master mode. SPSSR can be read by the CPU. Any writing to SPSSR by the CPU is ignored.



Bit:	7	6	5	4	3	2	1	0
	–	–	SPECM[1:0]		–	–	SPCP[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
5, 4	SPECM[1:0]	00	R	RSPI Error Command These bits indicate RSPI command registers 0 to 3 (SPCMD0 to SPCMD3) that are pointed to by command pointers (SPCP1 and SPCP0 bits) when an error is detected during sequence control by the RSPI. The RSPI updates the bits SPECM1 and SPECM0 only when an error is detected. If both the OVRF and MODF bits in the RSPI status register (SPSR) are 0 and there is no error, the values of the bits SPECM1 and SPECM0 have no meaning. For the RSPI's error detection function, see section 18.4.7, Error Detection. For the RSPI's sequence control, see section 18.4.9 (2), Master Mode Operation. 00: SPCMD0 01: SPCMD1 10: SPCMD2 11: SPCMD3
3, 2	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	SPCP[1:0]	000	R	RSPI Command Pointer During RSPI sequence control, these bits indicate RSPI command registers 0 to 3 (SPCMD0 to SPCMD3), which are currently pointed to by the pointers. For the RSPI's sequence control, see 18.4.9 (2), Master Mode Operation. 00: SPCMD0 01: SPCMD1 10: SPCMD2 11: SPCMD3

18.3.8 RSPI Bit Rate Register (SPBR)

SPBR sets the bit rate in master mode. SPBR can be read from or written to by the CPU. If the contents of SPBR are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed. When the RSPI is used in slave mode, the bit rate depends on the input clock regardless of the settings of SPBR and BRDV.

Bit:	7	6	5	4	3	2	1	0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bit rate is determined by combinations of SPBR settings and the bit settings in the BRDV1 and BRDV0 bits in the RSPI command registers (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 bit settings in the bits BRDV1 and BRDV0 (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f\,(P\phi)}{2 \times (N + 1) \times 2^n}$$

Table 18.3 shows examples of the relationship between the SPBR register and BRDV1 and BRDV0 bit settings.

Table 18.3 Relationship between SPBR and BRDV[1:0] Settings

SPBR (N)	BRDV[1:0] (n)	Division Ratio	Bit Rate				
			Pϕ = 16 MHz	Pϕ = 20 MHz	Pϕ = 32 MHz	Pϕ = 40 MHz	Pϕ = 50 MHz
0	0	2	8.0 Mbps	10.0 Mbps	—	—	—
1	0	4	4.0 Mbps	5.0 Mbps	8.0 Mbps	10.0 Mbps	12.5 Mbps
2	0	6	2.67 Mbps	3.3 Mbps	5.33 Mbps	6.67 Mbps	8.33 Mbps
3	0	8	2.0 Mbps	2.5 Mbps	4.0 Mbps	5.0 Mbps	6.25 Mbps
4	0	10	1.6 Mbps	2.0 Mbps	3.2 Mbps	4.0 Mbps	5.00 Mbps
5	0	12	1.33 Mbps	1.67 Mbps	2.67 Mbps	3.33 Mbps	4.17 Mbps
5	1	24	667 kbps	833 kbps	1.33 Mbps	1.67 Mbps	2.08 Mbps

SPBR (N)	BRDV[1:0] (n)	Division Ratio	Bit Rate				
			Pϕ = 16 MHz	Pϕ = 20 MHz	Pϕ = 32 MHz	Pϕ = 40 MHz	Pϕ = 50 MHz
5	2	48	333 kbps	417 kbps	667 kbps	833 kbps	1.04 Mbps
5	3	96	167 kbps	208 kbps	333 kbps	417 kbps	520 kbps
255	3	4096	3.9 kbps	4.9 kbps	7.8 kbps	9.8 kbps	10 kbps

[Legend]

—: Setting prohibited

18.3.9 RSPI Data Control Register (SPDCR)

RSPI sets the number of frames that can be stored in the SPDR register, specifies from which buffer of the SPDR register data should be read, and sets the access size, word or longword, for the SPDR register.

Up to 4 frames can be transmitted or received at a time upon transmission or reception activation according to the setting combinations of the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD), RSPI sequence length setting bits (SPSLN1 and SPSLN0) in the RSPI sequence control register (SPSCR), and frame count setting bits (SPFC1 and SPFC0) in the RSPI data control register (SPDCR).

SPDCR can be read from or written to by the CPU. If the contents of SPDCR are changed by the CPU while the RSPI function is enabled with the SPE bit in the RSPI control register (SPCR) set to 1, subsequent operations cannot be guaranteed.

Bit:	7	6	5	4	3	2	1	0
	–	–	SPLW	SPRDTD	–	–	SPFC[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5	SPLW	0	R/W	RWPI Longword/Word Access Setting Sets the access size for the RSPI data register (SPDR). When SPLW is set to 0, SPDR is accessed in word units. When SPLW is set to 1, SPDR is accessed in longword units. When SPLW is 0, the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD) should be set to 8 to 16 bits. If these bits are set to 20, 24, or 32 bits, operation cannot be guaranteed. 0: Word access to SPDR register 1: Longword access to SPDR register
4	SPRDTD	0	R/W	RSPI Receive/Transmit Data Select Selects whether data should be read from the receive buffer or transmit buffer of the RSPI data register (SPDR). When reading from the transmit buffer, most recently written value is read. Reading from the transmit buffer is allowed while the SPTEF bit in the RSPI status register (SPSR) is 1. 0: Read from receive buffer. 1: Read from transmit buffer (only when the SPTEF bit is 1).
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

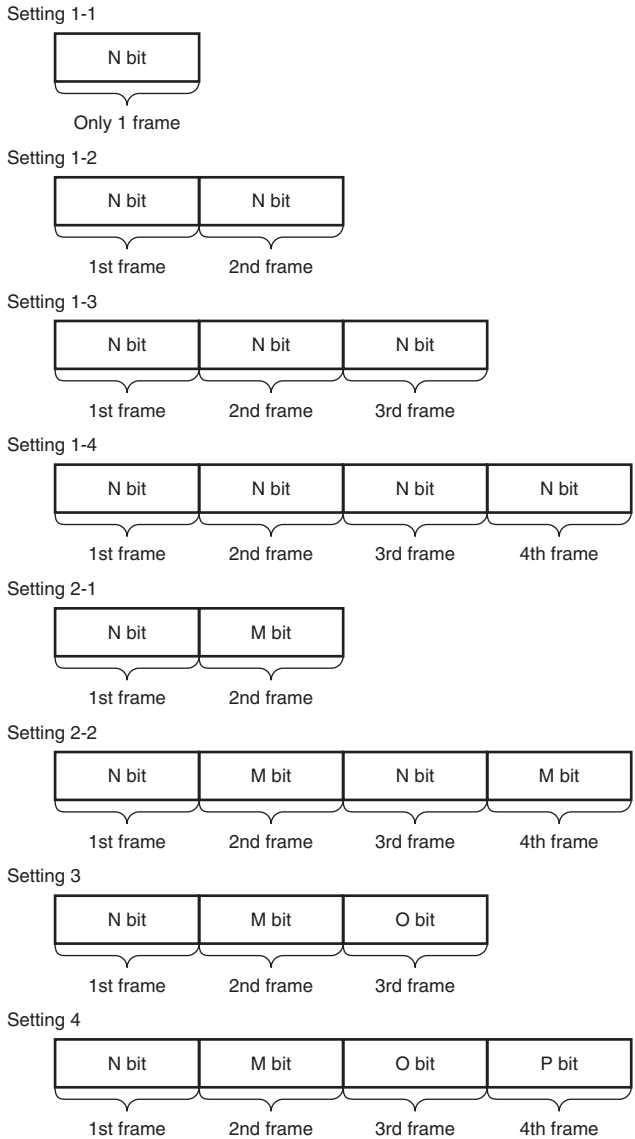
Bit	Bit Name	Initial Value	R/W	Description
1, 0	SPFC[1:0]	00	R/W	Frame Count Setting These bits specify the number of frames that can be stored in the SPDR register. Up to 4 frames can be transmitted or received at a time upon transmission or reception activation according to the setting combinations of the RSPI data length setting bits (SPB3 to SPB0) in the RSPI command register (SPCMD), RSPI sequence length setting bits (SPSLN1 and SPSLN0) in the RSPI sequence control register (SPSCR), and frame count setting bits (SPFC1 and SPFC0) in the RSPI data control register (SPDCR). These bits also specify the number of received data to set the RSPI receive buffer full flag in the RSPI status register (SPSR) and the number of remaining data to be transmitted to clear the RSPI transmit buffer empty flag in SPSR. Table 18.4 shows combination examples of the frame formats that can be stored in the SPDR register and the transmission/reception settings. If any setting other than those listed in table 18.4 is made, subsequent operations cannot be guaranteed.

Table 18.4 Combinations of Frame Count Setting Bits

Setting No.	SPB3 to SPB0	SPSLN1 and SPSLN0	SPFC1 and SPFC0	Number of Frames to Transfer	Number of Frames to Set SPRF to 1 or to Clear SPTEF to 0
1-1	N	00	00	1	1 frame
1-2	N	00	01	2	2 frames
1-3	N	00	10	3	3 frames
1-4	N	00	11	4	4 frames
2-1	N, M	01	01	2	2 frames
2-2	N, M	01	11	4	4 frames
3	N, M, O	10	10	3	3 frames
4	N, M, O, P	11	11	4	4 frames

[Legend] N, M, O, P: Data lengths that can be set with SPB3 to SPB0.

Data can be transferred or received at a time upon transmission or reception activation according to the setting combinations, 1-1 to 4, as follows:



18.3.10 RSPI Clock Delay Register (SPCKD)

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SCKDEN bit in the RSPI command register (SPCMD) is 1. SPCKD can be read from or written to by the CPU. If the contents of SPCKD are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

When using the RSPI in slave mode, set 000 in SCKDL[2:0].

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	SCKDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SCKDL[2:0]	000	R/W	RSPCK Delay Setting These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 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

18.3.11 SPI Slave Select Negation Delay Register (SSLND)

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. SSLND can be read from or written to by the CPU. If the contents of SSLND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

When using the RSPI in slave mode, set 000 in SLNDL[2:0].

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	SLNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SLNDL[2:0]	000	R/W	SSL Negation Delay Setting These bits set an SSL negation delay value when the RSPI is in master mode. 000: 1 RSPCK 001: 2 RSPCK 010: 3 RSPCK 011: 4 RSPCK 100: 5 RSPCK 101: 6 RSPCK 110: 7 RSPCK 111: 8 RSPCK

18.3.12 RSPI Next-Access Delay Register (SPND)

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPNDEN bit in the RSPI command register (SPCMD) is 1. SPND can be read from or written to by the CPU. If the contents of SPND are changed by the CPU while the MSTR and SPE bits in the RSPI control register (SPCR) are 1 with the RSPI function in master mode enabled, operation cannot be guaranteed.

When using the RSPI in slave mode, set 000 in SPNDL[2:0].

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	SPNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved The write value should always be 0. Otherwise, operation cannot be guaranteed.
2 to 0	SPNDL[2:0]	000	R/W	RSPI Next-Access Delay Setting These bits set a next-access delay when the SPNDEN bit in SPCMD is 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

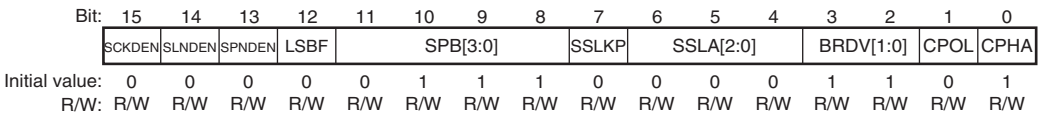
18.3.13 RSPI Command Register (SPCMD)

The RSPI has four RSPI command registers (SPCMD0 to SPCMD3). SPCMD0 to SPCMD3 are used to set a transfer format for the RSPI in master mode. 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 SPCMD3 according to the settings in bits SPSLN1 and SPSLN0 in the RSPI sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD.

SPCMD can be read from or written to by the CPU.

Set the SPCMD register before setting data to be transferred referencing the SPCMD settings while the SPTEF bit in the RSPI status register (SPSR) is 1.

SPCMD that is referenced by the RSPI in master mode can be checked by means of bits SPCP1 and SPCP0 in the RSPI sequence status register (SPSSR). When the RSPI function in slave mode is enabled, operation cannot be guaranteed if the value set in SPCMD0 is changed by the CPU.



Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	SCKDEN	0	R/W	RSPCK Delay Setting Enable Sets the period from the time the RSPI in master mode sets the SSL signal active until the RSPI oscillates RSPCK (RSPCK 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 RSPCK delay register (SPCKD) settings. To use the RSPI in slave mode, the SCKDEN bit should be set to 0. 0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay equal to SPCKD settings.

Bit	Bit Name	Initial Value	R/W	Description
14	SLNDEN	0	R/W	SSL Negation Delay Setting Enable 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. If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with slave select negation delay register (SSLND) settings. To use the RSPI in slave mode, the SLNDEN bit should be set to 0. 0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay equal to SSLND settings.
13	SPNDEN	0	R/W	RSPI Next-Access Delay Enable 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 + 2Pϕ. If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with RSPI next-access delay register (SPND) settings. To use the RSPI in slave mode, the SPNDEN bit should be set to 0. 0: A next-access delay of 1 RSPCK + 2 Pϕ 1: A next-access delay equal to SPND settings.
12	LSBF	0	R/W	RSPI LSB First Sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first. 0: MSB first 1: LSB first

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	SPB[3:0]	0111	R/W	SRPI Data Length Setting These bits set a transfer data length for the RSPI in master mode or slave mode. 0100 to 0111: 8 bits 1000: 9 bits 1001: 10 bits 1010: 11 bits 1011: 12 bits 1100: 13 bits 1101: 14 bits 1110: 15 bits 1111: 16 bits 0000: 20 bits 0001: 24 bits 0010 and 0011: 32 bits
7	SSLKP	0	R/W	SSL Signal Level Keeping When the RSPI in master mode performs a serial transfer, this 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. To use the RSPI in slave mode, the SSLKP bit should be set to 0. 0: Negates all SSL signals upon completion of transfer. 1: Keeps the SSL signal level from the end of the transfer until the beginning of the next access.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	SSLA[2:0]	000	R/W	SSL Signal Assertion Setting These bits control the SSL signal assertion when the RSPI performs serial transfers in master mode. Setting these bits controls the assertion for the signals SSL3 to SSL0. When an SSL signal is asserted, its polarity is determined by the set value in the corresponding SSLP (RSPI slave select polarity register). When the SSLA2 to SSLA0 bits are set to 000 or 1** in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as SSL0 acts as input). When the SSLA2 to SSLA0 bits are set to 1** in single-master mode, serial transfers are performed with all the SSL signals in the negated state as well. When using the RSPI in slave mode, set 000 in SSLA2 to SSLA0. 000: SSL0 001: SSL1 010: SSL2 011: SSL3 1xx: —

Bit	Bit Name	Initial Value	R/W	Description
3, 2	BRDV[1:0]	11	R/W	Bit Rate Division Setting These bits are used to determine the bit rate. A bit rate is determined by combinations of bits BRDV1 and BRDV 0 and the settings in the RSPI bit rate register (SPBR). The settings in SPBR determine the base bit rate. The settings in bits BRDV1 and BRDV0 are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. For SPCMD0 to SPCMD3, different BRDV1 and BRDV0 settings can be specified. This permits the execution of serial transfers at a different bit rate for each command. 00: Select the base bit rate 01: Select the base bit rate divided by 2 10: Select the base bit rate divided by 4 11: Select the base bit rate divided by 8
1	CPOL	0	R/W	RSPCK Polarity Setting Sets the RSPCK polarity of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules. 0: RSPCK = 0 when idle 1: RSPCK = 1 when idle
0	CPHA	1	R/W	RSPCK Phase Setting Sets the RSPCK phase of the RSPI in master or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules. 0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge

18.4 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.

18.4.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave (SPI), single-master (SPI), and multi-master (SPI), slave (clock synchronous), and master (clock synchronous) modes. A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in the RSPI control register (SPCR). Table 18.5 gives the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 18.5 Relationship between RSPI Modes and SPCR and Description of Each Mode

Item	Slave (SPI)	Single-Master (SPI)	Multi-Master (SPI)	Slave (Clock Synchronous)	Master (Clock Synchronous)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0, 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCK signal	Input	Output	Output/Hi-Z	Input	Output/Hi-Z
MOSI signal	Input	Output	Output/Hi-Z	Input	Output/Hi-Z
MISO signal	Output/Hi-Z	Input	Input	Output/Hi-Z	Input
SSL0 signal	Input	Output	Input	Hi-Z	Hi-Z
SSL1 to SSL3 signals	Hi-Z	Output	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	—	—
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator

Item	Slave (SPI)	Single-Master (SPI)	Multi-Master (SPI)	Slave (Clock Synchronous)	Master (Clock Synchronous)
Clock polarity	Two	Two	Two	Two	Two
Clock phase	Two	Two	Two	One (CPHA = 1)	One (CPHA = 1)
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 starting method	SSL input active or RSPCK oscillation	Writing to transmit buffer when SPTEF = 1	Writing to transmit buffer when SPTEF = 1	RSPCK oscillation	Writing to transmit buffer when SPTEF = 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported	Supported	Supported	Supported	Supported
Receive buffer full detection	Supported	Supported	Supported	Supported	Supported
Overrun error detection	Supported	Supported	Supported	Supported	Supported
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

18.4.2 Controlling RSPI Pins

According to the MSTR, MODFEN and SPMS bits in the RSPI control register (SPCR) and the SPOM bit in the RSPI pin control register (SPPCR), the RSPI can automatically switch pin directions and output modes. Table 18.6 shows the relationship between pin states and bit settings.

Table 18.6 Relationship between Pin States and Bit Settings

		Pin State*1	
Mode	**Pin**	**SPOM = 0**	**SPOM = 1**
Single-master mode (SPI) (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) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCK*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL3*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO	Input	Input
Slave mode (SPI) (MSTR = 0, SPMS = 0)	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL3	Hi-Z	Hi-Z
	MOSI	Input	Input
	MISO*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master (clock synchronous) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3*4	Hi-Z	Hi-Z
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input

		Pin State*1	
Mode	**Pin**	**SPOM = 0**	**SPOM = 1**
Slave (clock synchronous) (MSTR = 0, SPMS = 1)	RSPCK	Input	Input
	SSL0 to SSL3*4	Hi-Z	Hi-Z
	MOSI	Input	Input
	MISO	CMOS output	Open-drain output

Notes: 1. RSPI settings are not reflected to the multi-function pins for which the RSPI function is not applied.
2. When SSL0 is at the active level, the pin state is Hi-Z.
3. When SSL0 is at the active level or the SPE bit in SPCR is 0, the pin state is Hi-Z.
4. SSL0 to SSL3 can be used as the IO ports in clock synchronous mode.

The RSPI in single-master (SPI) and multi-master (SPI) modes determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to the settings of the MOIFE and MOIFV bits in SPPCR as shown in table 18.7.

Table 18.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE	**MOIFV**	**MOSI Signal Value during SSL Negation Period***
0	0, 1	Final data from previous transfer
1	0	Always 0
1	1	Always 1

Note: * The SSL negation period includes the SSL retention period during a burst transfer.

18.4.3 RSPI System Configuration Example

(1) Single Master/Single Slave (with This LSI Acting as Master)

Figure 18.2 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 outputs of this LSI (master) are not used. The SSL input of the RSPI slave is fixed to 0, and the RSPI slave is always maintained in a select state. In the transfer format corresponding to the case where the CPHA bit in the RSPI control register (SPCR) 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.

This LSI (master) always drives the RSPCK and MOSI signals. The RSPI slave always drives the MISO signal.

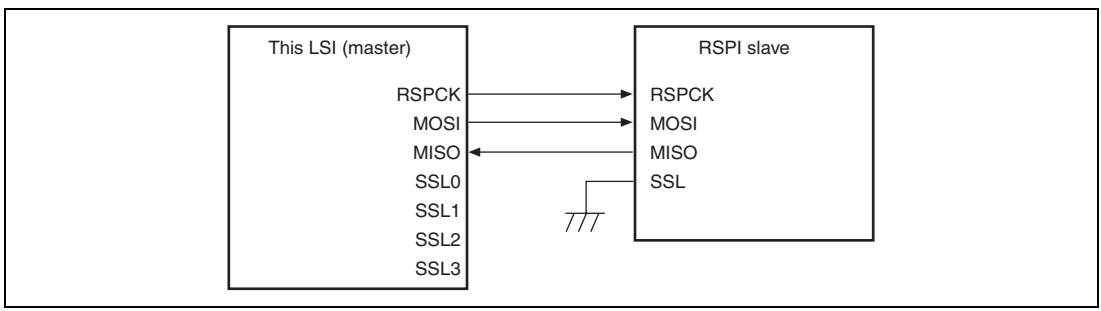


Figure 18.2 Single-Master/Single-Slave Configuration Example (This LSI = Master)

(2) Single Master/Single Slave (with This LSI Acting as Slave)

Figure 18.3 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 signals. This LSI (slave) always drives the MISO signal*.

In the single-slave configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, the SSL0 input of this LSI (slave) is fixed to 0, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (figure 18.4).

Note: * When SSL0 is at the active level, the pin state becomes Hi-Z.

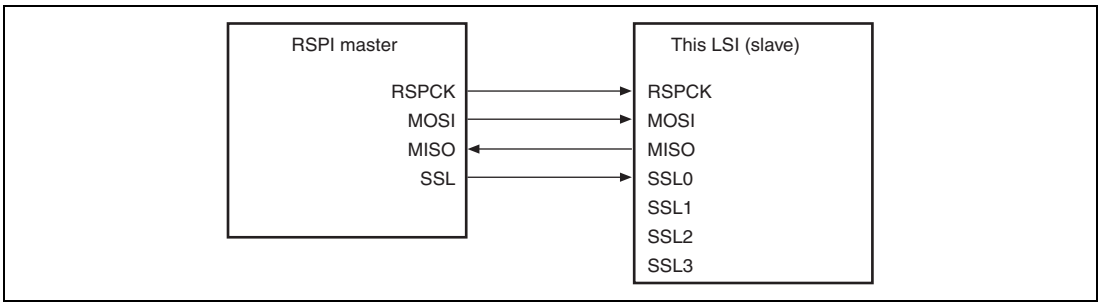


Figure 18.3 Single-Master/Single-Slave Configuration Example (This LSI = Slave)

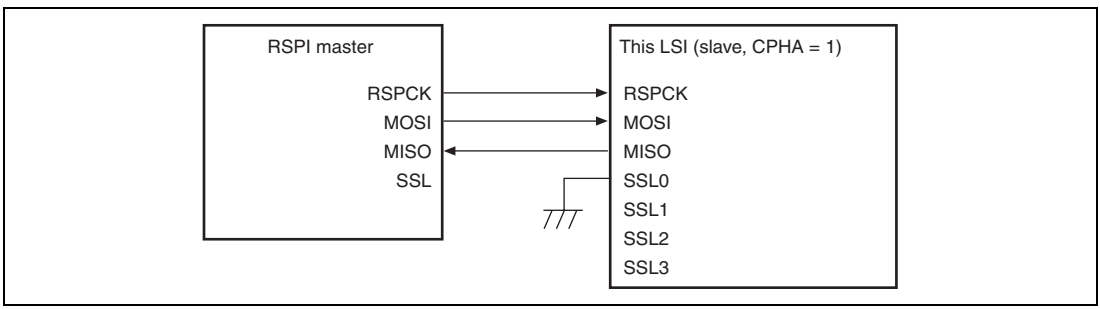


Figure 18.4 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

(3) Single Master/Multi-Slave (with This LSI Acting as Master)

Figure 18.5 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 18.5, 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 the RSPCK, MOSI, and SSL0 to SSL3 signals. Of the RSPI slave 0 to RSPI slave 3, the slave that receives 0 into the SSL input drives the MISO signal.

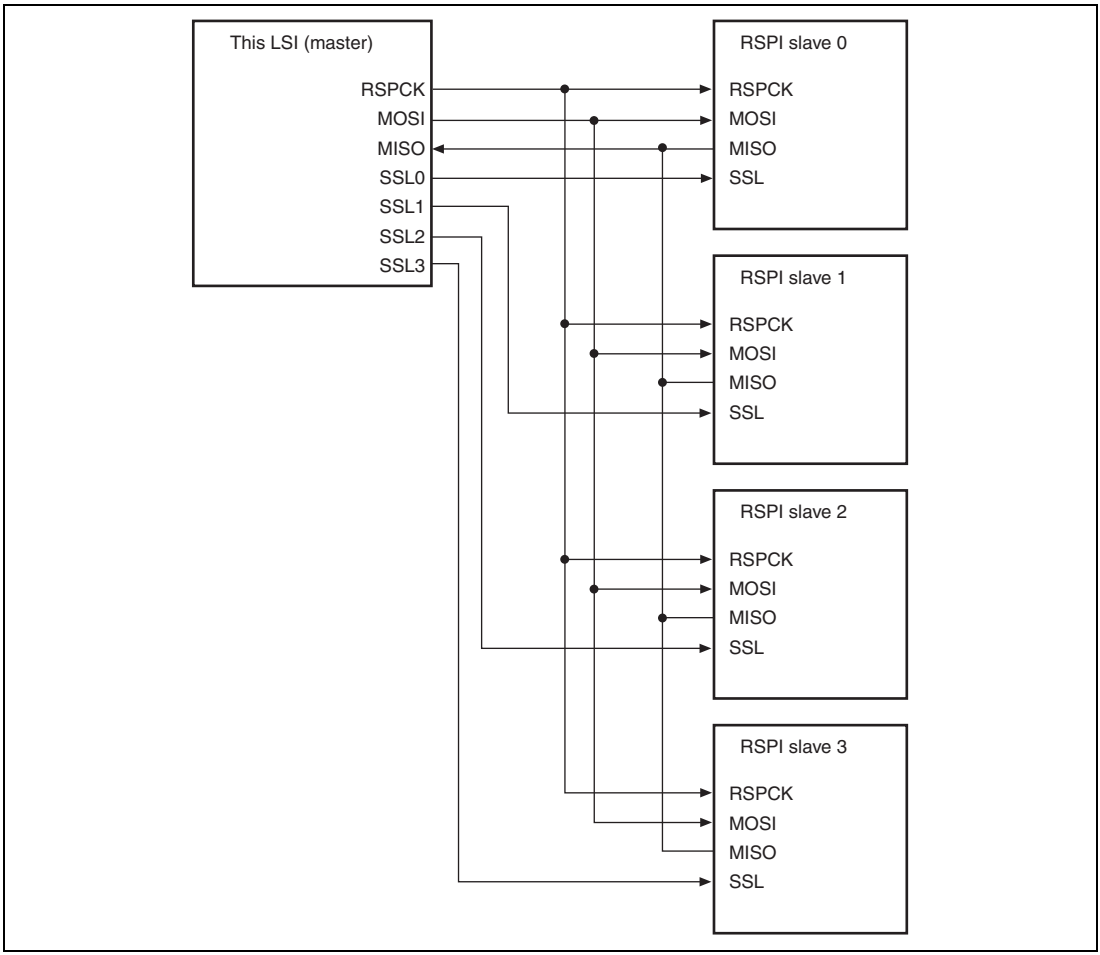


Figure 18.5 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

(4) Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 18.6 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of figure 18.6, the RSPI system is comprised of an RSPI master and these 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 these LSIs (slave X and slave Y). The MISO outputs of these 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 the RSPCK, MOSI, SSLX, and SSLY signals. Of these LSIs (slave X and slave Y), the slave that receives low level input into the SSL0 input drives the MISO signal.

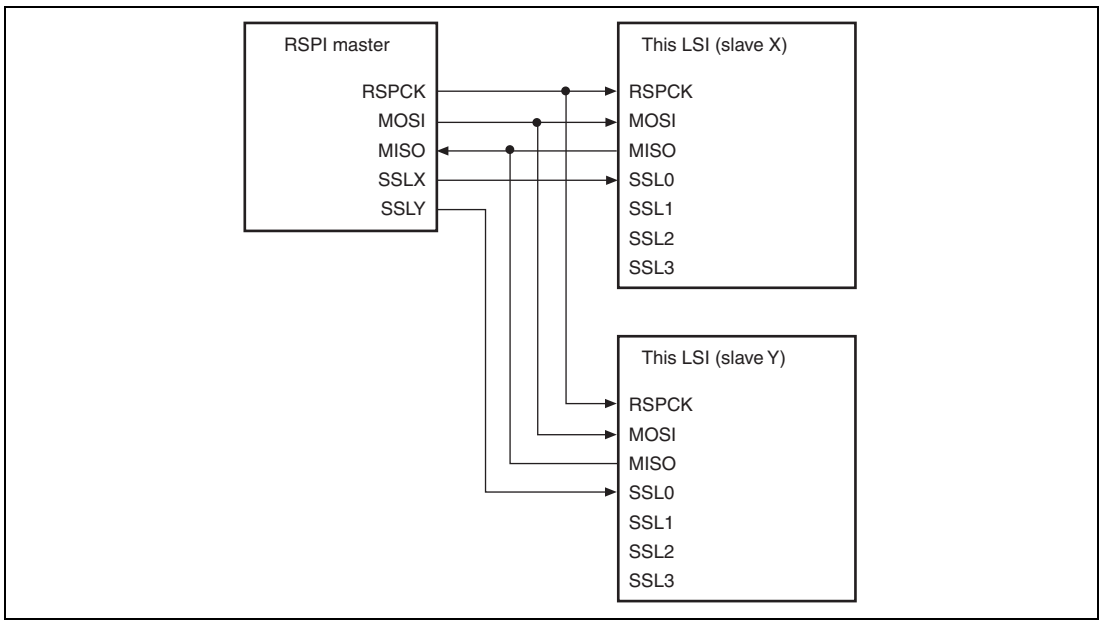


Figure 18.6 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

(5) Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 18.7 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of figure 18.7, the RSPI system is comprised of these two LSIs (master X, master Y) and two RSPI slaves (RSPI slave 1, RSPI slave 2).

The RSPCK and MOSI outputs of this LSI (master X, 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 this LSI (master X, 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 this LSI (master X, 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 output SSL3 of this LSI is not required.

This LSI drives the RSPCK, MOSI, SSL1, and SSL2 signals 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 Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives 0 into the SSL input drives the MISO signal.

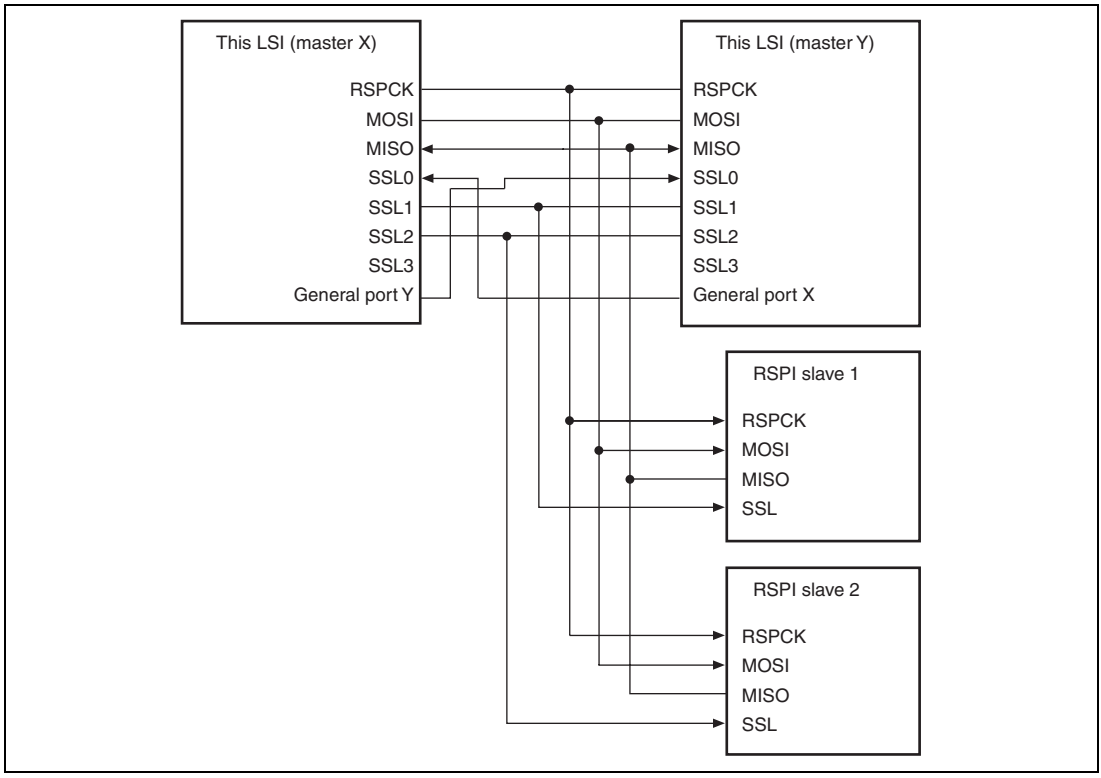


Figure 18.7 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

(6) Master (Clock Synchronous)/Slave (Clock Synchronous) (with This LSI Acting as Master)

Figure 18.8 shows a master (clock synchronous)/slave (clock synchronous) RSPI system configuration example when this LSI is used as a master. In the master (clock synchronous)/slave (clock synchronous) configuration, the SSL0 to SSL3 outputs of this LSI (master) are not used.

This LSI (master) always drives the RSPCK and MOSI signals. The RSPI slave always drives the MISO signal.

Only in the single-master configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, this LSI (master) can execute serial transfer.

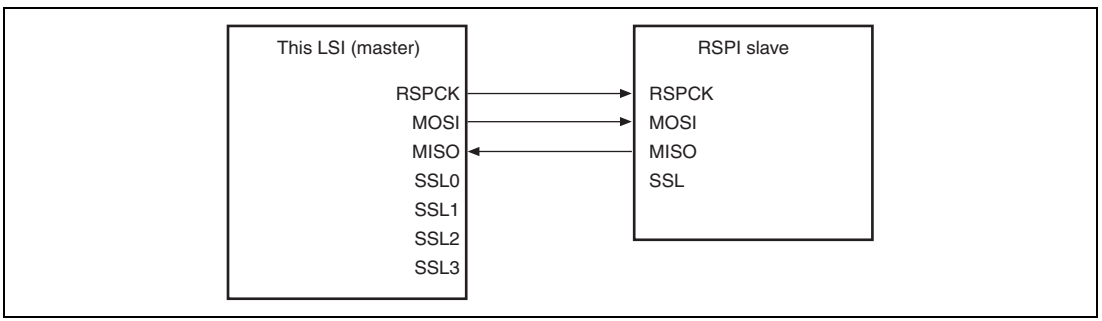


Figure 18.8 Master (Clock Synchronous)/Slave (Clock Synchronous) Configuration Example (This LSI = Master)

(7) Master (Clock Synchronous)/Slave (Clock Synchronous) (with This LSI = Slave)

Figure 18.9 shows a master (clock synchronous)/slave (clock synchronous) RSPI system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, this LSI always drives the MISO signal, and the RSPI master always drives the RSPCK and MOSI signals.

Only in the single-slave configuration in which the CPHA bit in the RSPI command register (SPCMD) is set to 1, this LSI (slave) can execute serial transfer.

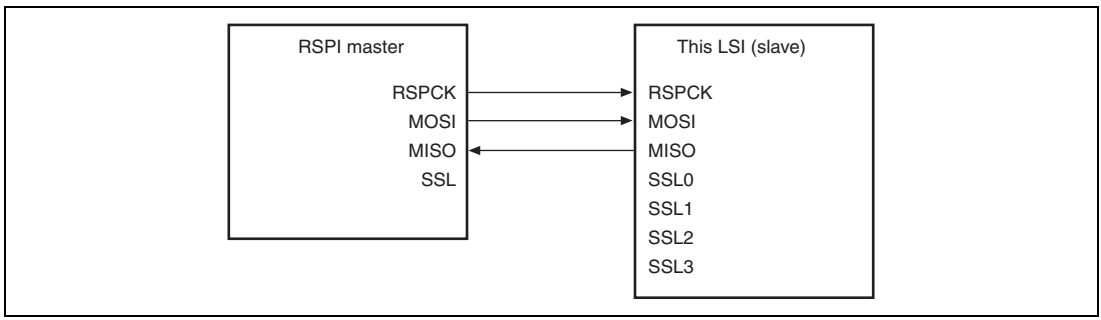


Figure 18.9 Master (Clock Synchronous)/Slave (Clock Synchronous) Configuration Example (This LSI = Slave, CPHA = 1)

18.4.4 Transfer Format

(1) CPHA = 0

Figure 18.10 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 0. Note that clock synchronous operation (with the SPMS bit in the RSPI control register (SPCR) set to 1) is not guaranteed when the CPHA bit is set to 0. In figure 18.10, 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 18.4.2, Controlling RSPI Pins.

When the CPHA bit is 0, the output of valid data to the MOSI signal and the driving of valid data to the MISO signal commence at an SSL signal assertion timing. The first RSPCK signal change timing that occurs after the SSL signal assertion becomes the first transfer data fetching timing. After this timing, data is sampled at every RSPCK cycle. The change timing for MOSI and MISO signals is always 1/2 RSPCK cycle after the transfer data fetch timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the cessation 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 18.4.9, SPI Operation.

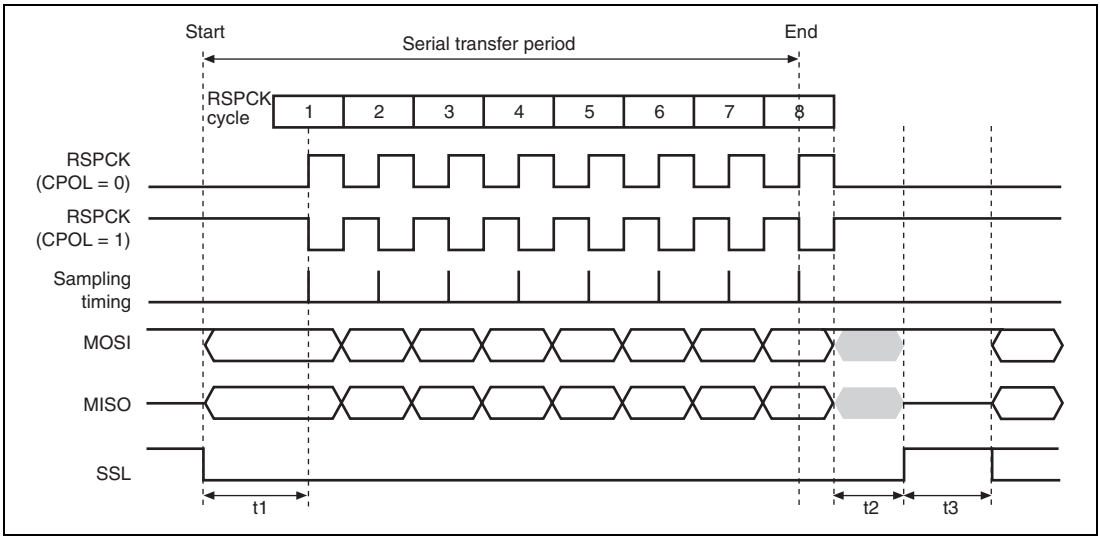


Figure 18.10 RSPI Transfer Format (CPHA = 0)

(2) CPHA = 1

Figure 18.11 shows an example transfer format for the serial transfer of 8-bit data when the CPHA bit in the RSPI command register (SPCMD) is 1. Note that when the SPMS bit in the RSPI control register (SPCR) is 1, the SSL signal is not used and only the RSPCK, MOSI, and MISO signals are used for communication. In figure 18.11, 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 RSPI mode (master or slave). For details, see section 18.4.2, Controlling RSPI Pins.

When the CPHA bit is 1, the driving of invalid data to the MISO signals commences at an SSL signal assertion timing. The driving 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 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycle after the data update timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect 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 18.4.9, SPI Operation.

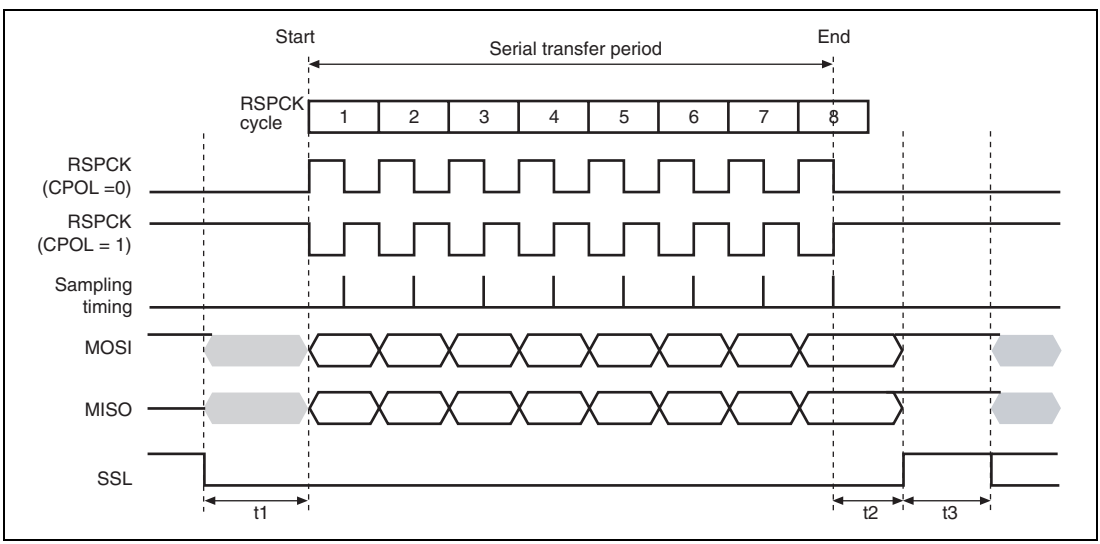


Figure 18.11 RSPI Transfer Format (CPHA = 1)

18.4.5 Data Format

The RSPI's data format depends on the settings in the RSPI command register (SPCMD). Irrespective of MSB/LSB first, the RSPI treats the assigned data length of data from the LSB of the RSPI data register (SPDR) as transfer data.

(1) MSB First Transfer (32-Bit Data)

Figure 18.12 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 32-bit MSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 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 cycle required for the serial transfer of 32 bits has passed, data R31 to R00 is stored in the shift register. In this state, 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 the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R31 to R00 is shifted out from the shift register.

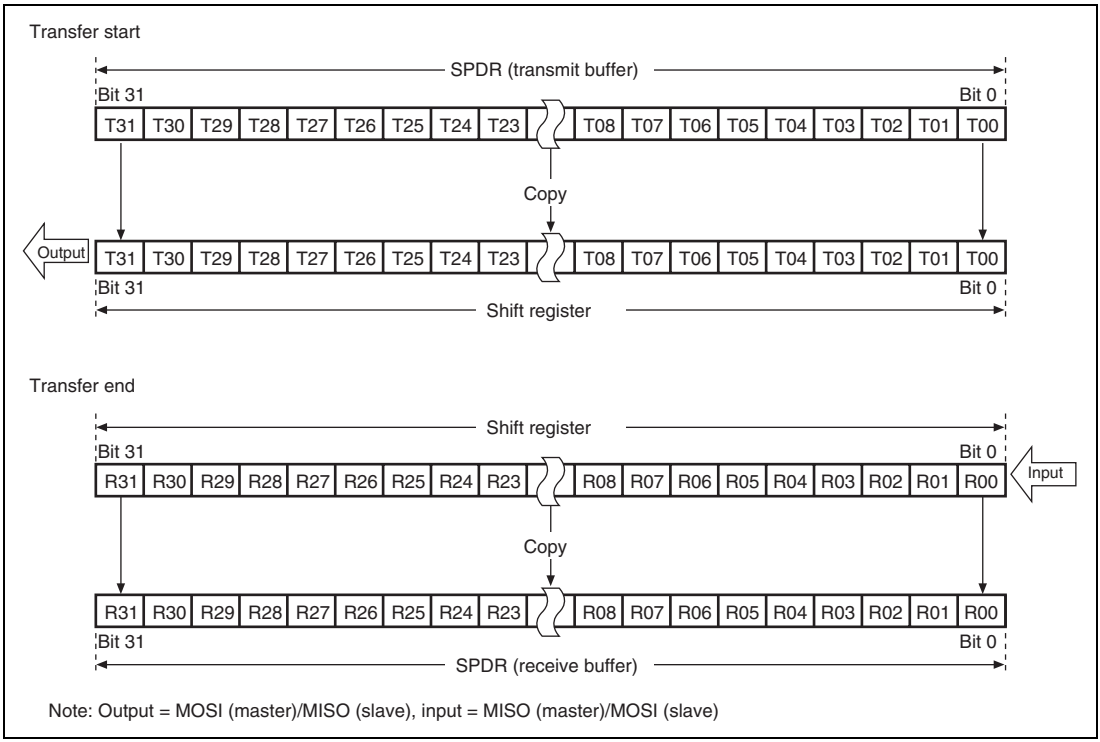


Figure 18.12 MSB First Transfer (32-Bit Data)

(2) MSB First Transfer (24-Bit Data)

Figure 18.13 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 24-bit data length MSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 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 bit 23 of the shift register, and shifts in the data from the LSB (bit 0) of the shift register. When the RSPCK cycle required for the serial transfer of 24 bits has 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 in the shift register. In this state, 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 the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R23 to R00 is shifted out from the shift register.

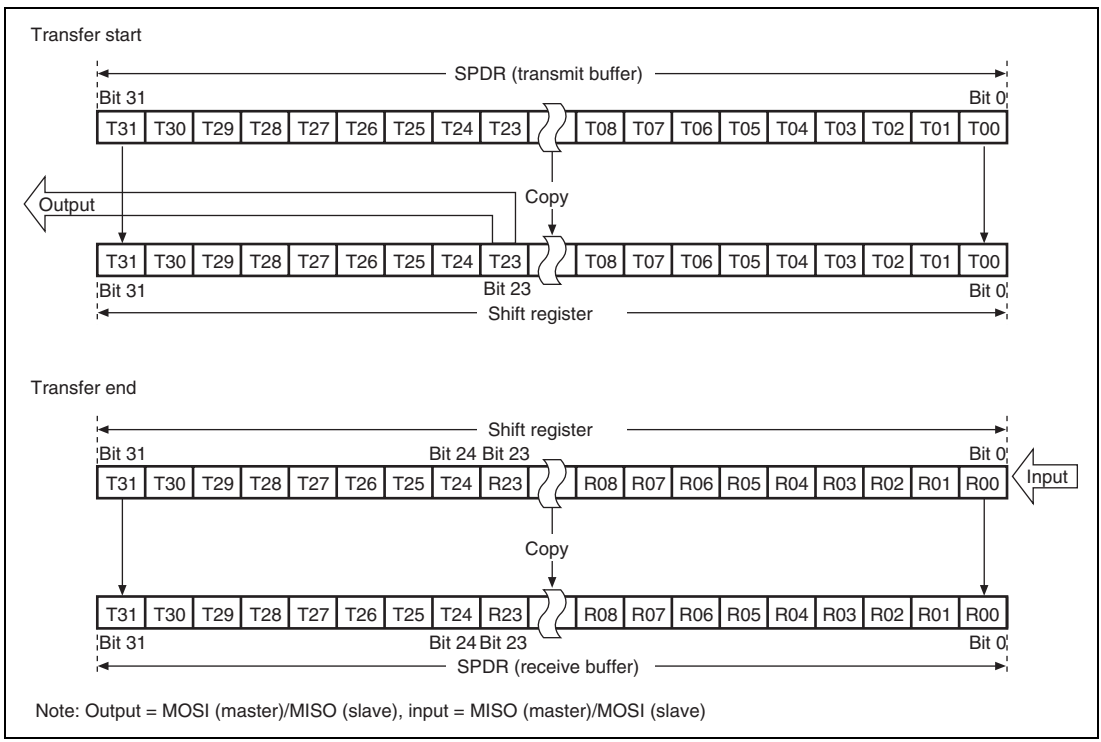


Figure 18.13 MSB First Transfer (24-Bit Data)

(3) LSB First Transfer (32-Bit Data)

Figure 18.14 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 32-bit data length LSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 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 cycle required for the serial transfer of 32 bits has passed, data R00 to R31 is stored in the shift register. In this state, 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 the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R00 to R31 is shifted out from the shift register.

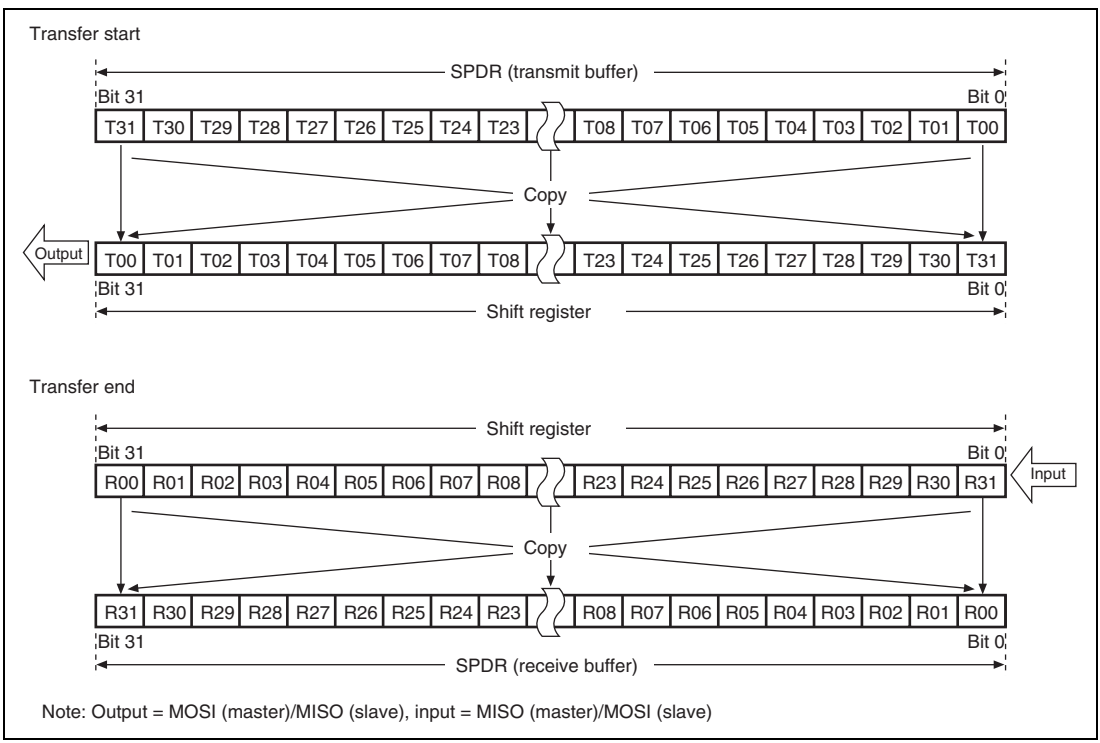


Figure 18.14 LSB First Transfer (32-Bit Data)

(4) LSB First Transfer (24-Bit Data)

Figure 18.15 shows the operation of the RSPI data register (SPDR) and the shift register when the RSPI performs a 24-bit data length LSB-first data transfer.

The CPU or the DTC/DMAC writes T31 to T00 to the transmit buffer of SPDR. If the SPTEF bit in the RSPI status register (SPSR) is 0 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 cycle required for the serial transfer of 24 bits has 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, 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 the CPU or the DTC/DMAC writes to the transmit buffer of SPDR, received data R00 to R23 is shifted out from the shift register.

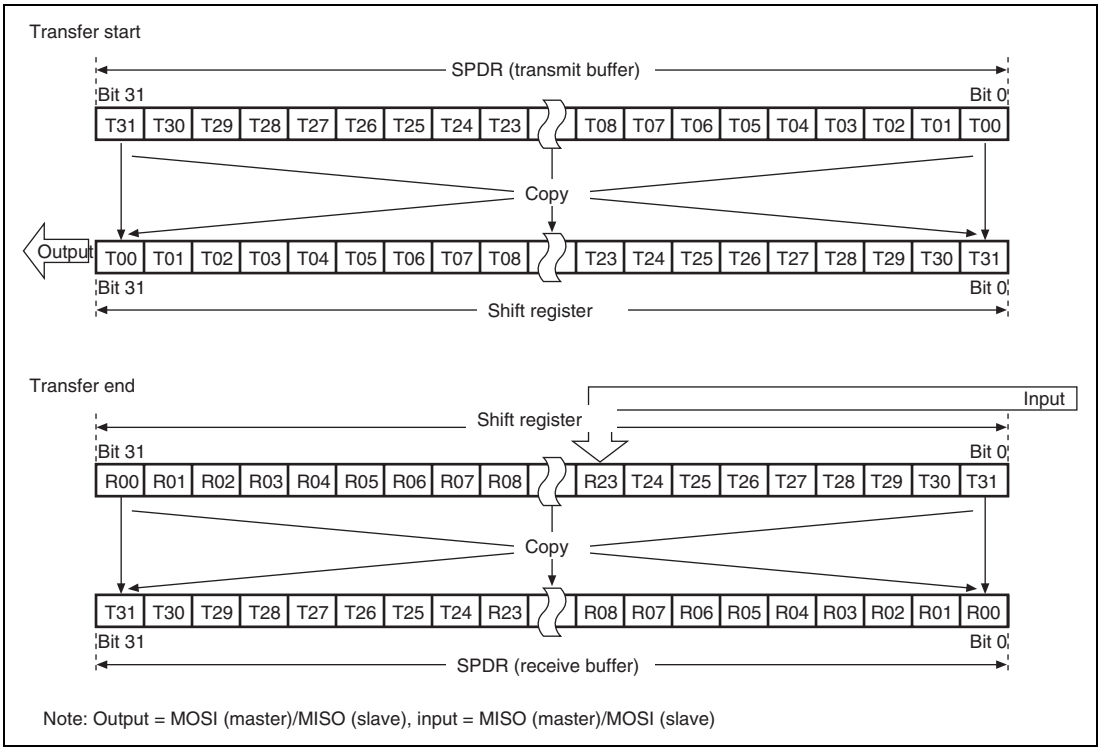


Figure 18.15 LSB First Transfer (24-Bit Data)

18.4.6 Transmit Buffer Empty/Receive Buffer Full Flags

Figure 18.16 shows an example of operation of the RSPI transmit buffer empty flag (SPTEF) and the RSPI receive buffer full flag in the RSPI status register (SPSR). The SPDR access depicted in figure 18.16 indicates the condition of access from the DTC/DMAC to the RSPI data register (SPDR), where I denotes an idle cycle, W a write cycle, and R a read cycle. In this example in figure 18.16, the RSPI executes an 8-bit serial transfer with the SPFC[1:0] bits in the RSPI data control register (SPDCR) set to 00, the CPHA bit in the RSPI command register (SPDR) set to 1, and the CPOL bit in SPDR set to 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

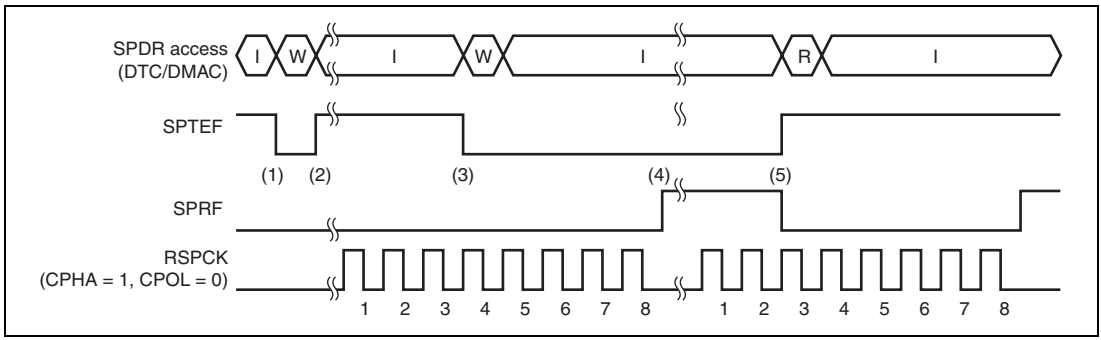


Figure 18.16 SPTEF and SPRF Bit Operation Example

The operation of the flags at timings shown in steps (1) to (5) in the figure is described below.

1. When the DTC/DMAC writes transmit data to SPDR when the transmit buffer of SPDR is empty, the RSPI sets the SPTEF bit to 0, and writes data to the transmit buffer, with no change in the SPRF flag.
2. If the shift register is empty, the RSPI sets the SPTEF bit to 1, and copies the data in the transmit buffer to the shift register, with no change in the SPRF flag. How a serial transfer is started depends on the mode of the RSPI. For details, see section 18.4.9, SPI Operation, and section 18.4.10, Clock Synchronous Operation.
3. When the DTC/DMAC writes transmit data to SPDR with the transmit buffer of SPDR being empty, the RSPI sets the SPTEF bit to 1, and writes data to the transmit buffer, while the SPRF flag remains unchanged. 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 sets the SPRF bit to 1, and copies the receive data in the shift register to the receive buffer. Because the shift register becomes empty upon completion of serial transfer, if the transmit buffer was full before the serial transfer ended, the RSPI sets the SPTEF bit to 1, and 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, and as a result data transfer from the transmit buffer to the shift register is enabled.
5. When the DTC/DMAC reads SPDR with the receive buffer being full, the RSPI sets the SPRF bit to 0, and sends the data in the receive buffer to the bus inside the chip.

If the CPU or the DTC/DMAC writes to SPDR when the SPTEF bit is 0, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure that the SPTEF bit is 1. That the SPTEF bit is 1 can be checked by reading SPSR or by using an RSPI transmit interrupt. 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), the SPTEF bit is initialized to 1. For this reason, setting the SPTIE bit to 1 when the RSPI is disabled generates an RSPI transmit interrupt.

When serial transfer ends with the SPRF bit being 1, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 18.4.7, Error Detection). To prevent a receive data overrun error, set the SPRF bit to 0 before the serial transfer ends. That the SPRF bit is 1 can be checked by either reading SPSR or by using an RSPI receive interrupt. To use an RSPI receive interrupt, set the SPRIE bit in SPCR to 1.

18.4.7 Error Detection

In the normal RSPI serial transfer, the data written from the RSPI data register (SPDR) to the transmit buffer by either the CPU or the DTC is serially transmitted, and either the CPU or the DTC/DMAC can read the serially received data from the receive buffer of SPDR. If access is made to SPDR by either the CPU or the DTC, depending on the status of the transmit buffer/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 or a mode fault error. Table 18.8 shows the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 18.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
A	Either the CPU or the DTC/DMAC writes to SPDR when the transmit buffer is full.	Retains the contents of the transmit buffer. 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	Either the CPU or the DTC/DMAC reads from SPDR when the receive buffer is empty.	Previously received serial data is output to the CPU or the DMAC.	None
D	Serial transfer terminates when the receive buffer is full.	Retains the contents of the receive buffer. Missing serial receive data.	Overrun error
E	The SSL0 input signal is asserted when the serial transfer is idle in multi-master mode.	RSPI disabled. Driving of the RSPCK, MOSI, and SSL1 to SSL3 output signals stopped.	Mode fault error

	Occurrence Condition	RSPI Operation	Error Detection
F	The SSL0 input signal is asserted during serial transfer in multi-master mode.	Serial transfer suspended. Missing send/receive data. Driving of the RSPCK, MOSI, and SSL1 to SSL3 output signals stopped. RSPI disabled.	Mode fault error
G	The SSL0 input signal is negated during serial transfer in slave mode.	Serial transfer suspended. Missing send/receive data. Driving of the MISO output signal stopped. RSPI disabled.	Mode fault error

On operation A shown in table 18.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR by the CPU or the DTC/DMAC, write operations to SPDR should be executed when the SPTEF bit in the RSPI status register (SPSR) is 1.

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. Notice that the received data from the previous serial transfer is retained in the receive buffer of SPDR, and thus it can be correctly read by the CPU or the DTC/DMAC (if SPDR is not read before the end of the serial transfer, an overrun error may result).

Similarly, the RSPI does not detect an error on operation C. To prevent the CPU or the DTC/DMAC from reading extraneous data, SPDR read operation should be executed when the SPRF bit in SPSR is 1.

An overrun error shown in D is described in section 18.4.7 (1), Overrun Error. A mode fault error shown in E to G is described in section 18.4.7 (2), Mode Fault Error. On operations of the SPTEF and SPRF bits in SPSR, see section 18.4.6, Transmit Buffer Empty/Receive Buffer Full Flags.

(1) Overrun Error

If serial transfer ends when the receive buffer of the RSPI data register (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 reset the OVRF bit in SPSR to 0, either execute a system reset, or write a 0 to the OVRF bit after the CPU has read SPSR with the OVRF bit set to 1.

Figure 18.17 shows an example of operation of the SPRF and OVRF bits in SPSR. The SPSR access depicted in figure 18.17 indicates the condition of access from the CPU to SPSR, and from the DTC/DMAC to SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of figure 18.17, the RSPI performs an 8-bit serial transfer in which the CPHA bit in the RSPI command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

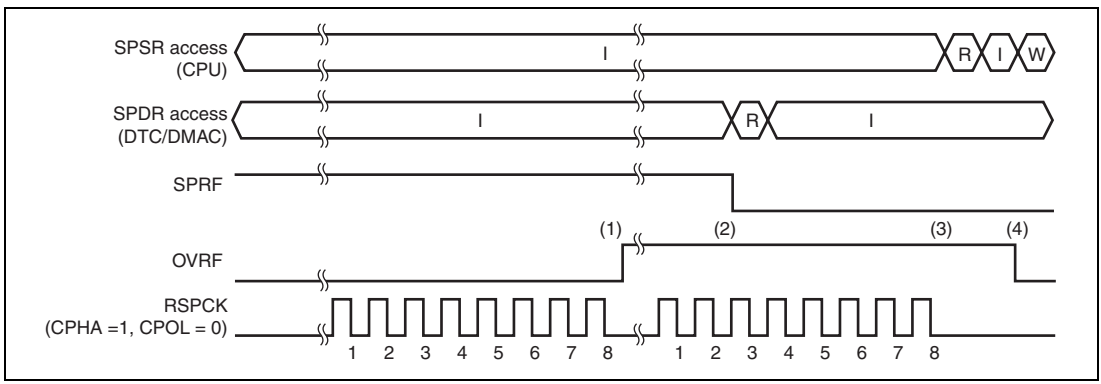


Figure 18.17 SPRF and OVRF Bit Operation Example

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 SPRF bit being 1 (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. In master mode, the RSPI copies the value of the pointer to the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR).
2. When the DTC/DMAC reads SPDR, the RSPI sets the SPRF bit to 0, and 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 terminates with the OVRF bit being 1 (an overrun error), the RSPI keeps the SPRF bit at 0 and does not update it. Likewise, the RSPI does not copy the data in the shift register to the receive buffer. When in master mode, the RSPI does not update bits SPECM1 and SPECM0 of SPSSR. If, in an overrun error state, 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 is enabled from the transmit buffer to the shift register.
4. If the CPU writes a 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 run in master mode, the pointer value to SPCMD can be checked by reading bits SPECM2 to SPECM0 of SPSSR.

If an overrun error occurs and the OVRF bit is set to 1, normal reception operations cannot be performed until such time as the OVRF bit is cleared. The OVRF bit is cleared to 0 under the following conditions:

- After reading SPSR in a condition in which the OVRF bit is set to 1, the CPU writes a 0 to the OVRF bit.
- System reset

(2) Mode Fault Error

The RSPI operates in multi-master mode when the MSTR bit is 1, the SPMS bit is 0 and the MODFEN bit is 1 in the RSPI control register (SPCR). 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 the RSPI command register (SPCMD) to bits SPECM2 to SPECM0 in the RSPI sequence status register (SPSSR). 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 is 1 and the SPMS bit is 0 in the RSPI in slave mode 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 the driving of output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the RSPI function is disabled (see section 18.4.8, Initializing RSPI). In multi-master configuration, it is possible to release the master right by using a mode fault error to stop the driving of output signals and the RSPI function.

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, one can read bits SPECM2 to SPECM0 of SPSSR to verify the value of the pointer to SPCMD when an error occurs.

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 conditions:

- After reading SPSR in a condition where the MODF bit has turned 1, the CPU writes a 0 to the MODF bit.
- System reset

18.4.8 Initializing RSPI

If the CPU writes a 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 a part of the module function. If a system reset occurs, the RSPI initializes all of the module function. An explanation follows of initialization by the clearing of the SPE bit and initialization by a system reset.

(1) Initialization by Clearing 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 only in slave mode (Hi-Z)
- Initializing the internal state of the RSPI
- Initializing the SPTEF bit in the RSPI status register (SPSR)

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 SPRF, OVRF, and MODF bits 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 SPTEF bit in SPSR is initialized to 1. 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, a 0 should be written to the SPTIE bit simultaneously with the writing of a 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 a 0 to the SPTIE bit.

(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 (1), Initialization by Clearing SPE Bit.

18.4.9 SPI Operation

(1) Slave Mode Operation

(1-1) Starting a Serial Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) 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, the asserting 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 MSO 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 intact, in the full state.

Irrespective of CPHA bit settings, the timing at which the RSPI starts driving MISO output signals is the SSL0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on CPHA bit settings.

For details on the RSPI transfer format, see section 18.4.4, 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).

(1-2) Terminating a Serial Transfer

Irrespective of the CPHA bit in RSPI command register 0 (SPCMD0), the RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the SPRF bit in the RSPI status register (SPSR) is 0 and 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). Irrespective of the value of the SPRF bit, 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 18.4.7, Error Detection).

The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. The polarity

of the SSL0 input signal depends on the setting in the SSL0P bit in the RSPI slave select polarity register (SSLP). For details on the RSPI transfer format, see section 18.4.4, Transfer Format.

(1-3) Notes on Single-Slave Operations

If the CPHA bit in RSPI command register 0(SPCMD0) 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 18.4 as an example, if the RSPI is used in single-slave mode, the SSL0 signal is always fixed at active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute send/receive operation by the RSPI in a configuration in which the SSL0 input signal is fixed at 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.

(1-4) Burst Transfer

If the CPHA bit in RSPI command register 0 (SPCMD0) 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 access.

If the CPHA bit is 0, for the reason given in (1-3), Notes on Single-Slave Operations, second and subsequent serial transfers during the burst transfer cannot be executed correctly.

(1-5) Initialization Flowchart

Figure 18.18 shows an example of initialization flowchart for using the RSPI in slave mode during SPI operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

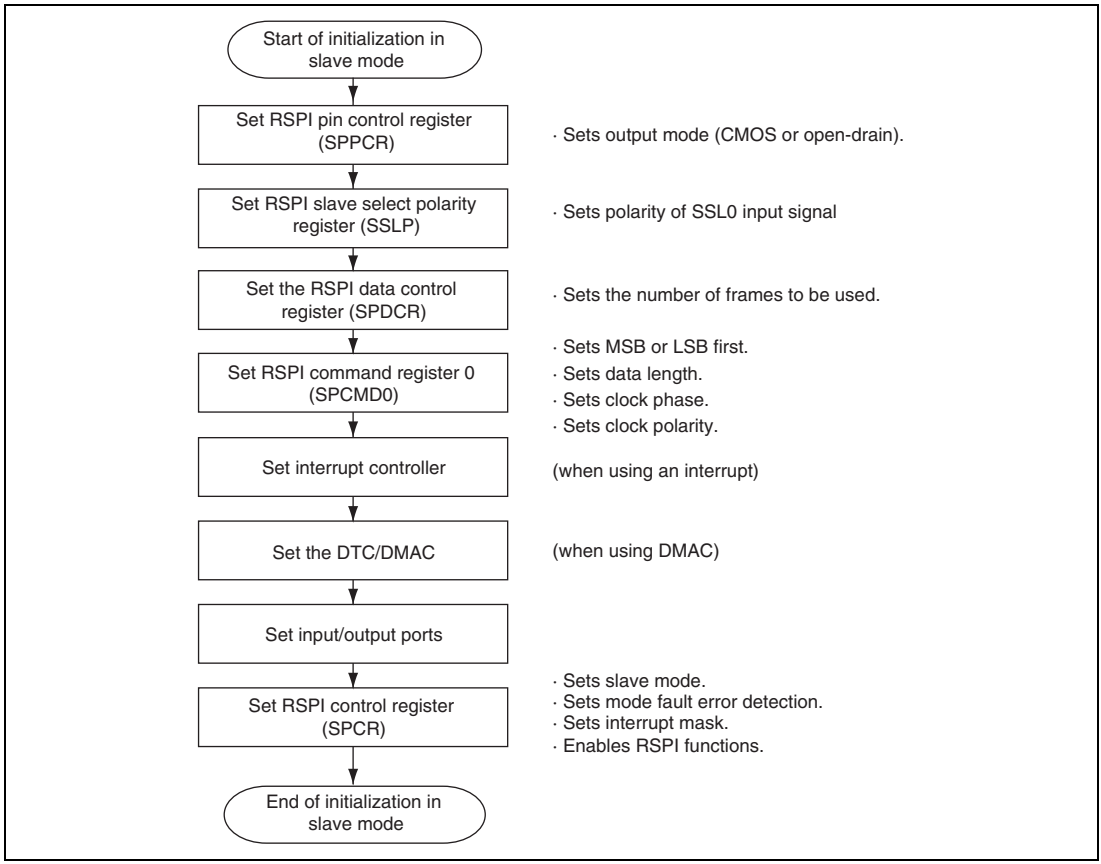


Figure 18.18 Example of Initialization Flowchart in Slave Mode

(1-6) Transfer Operation Flowchart (CPHA = 0)

Figure 18.19 shows an example of transfer operation flowchart for using the RSPI in slave mode during SPI operation, when the CPHA bit in RSPI command register 0 (SPCMD0) is 0.

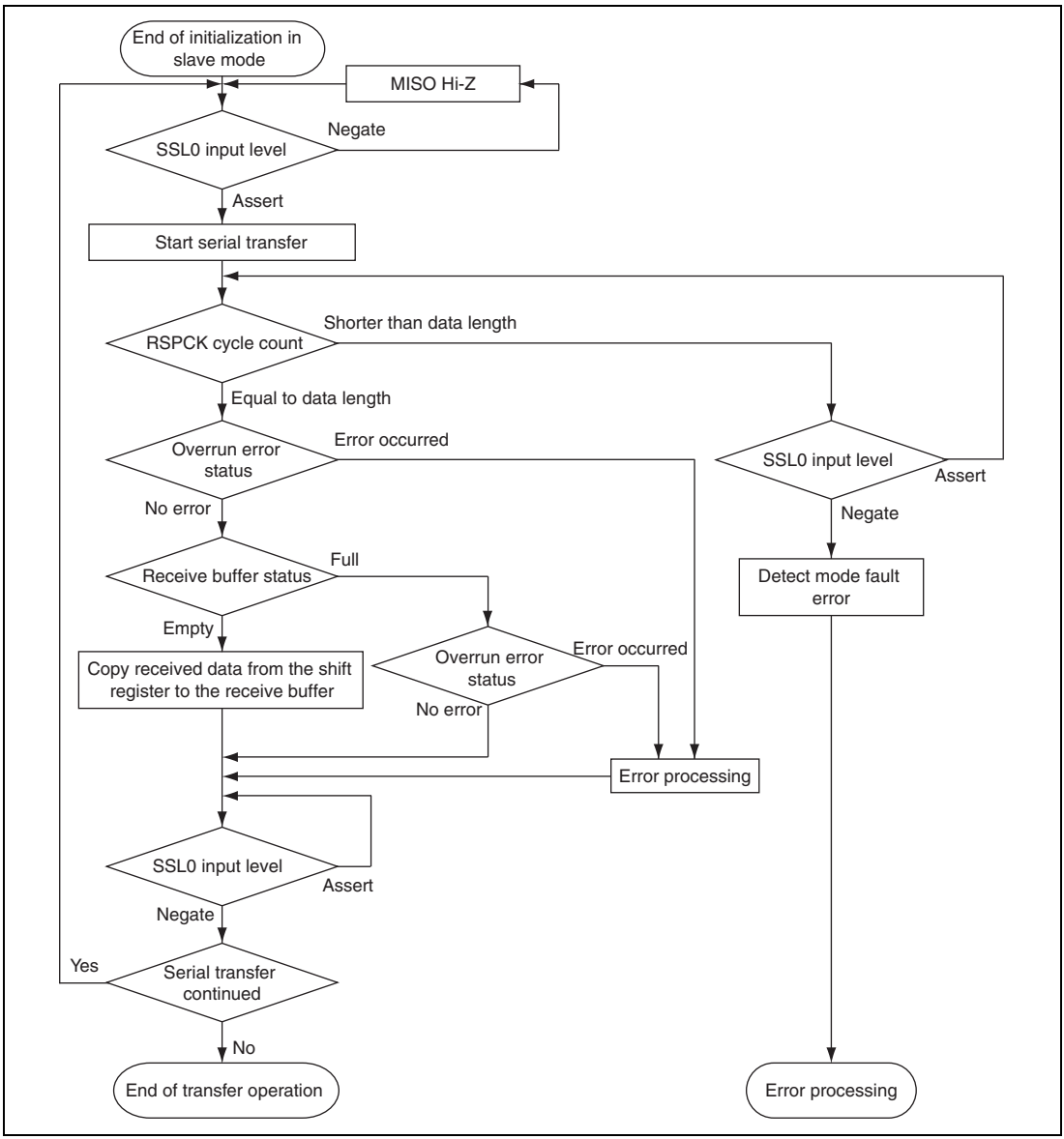


Figure 18.19 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 0)

(1-7) Transfer Operation Flowchart (CPHA = 1)

Figure 18.20 shows an example of transfer operation flowchart for using the RSPI in slave mode during SPI operation, when the CPHA bit in RSPI command register 0 (SPCMD0) is 1.

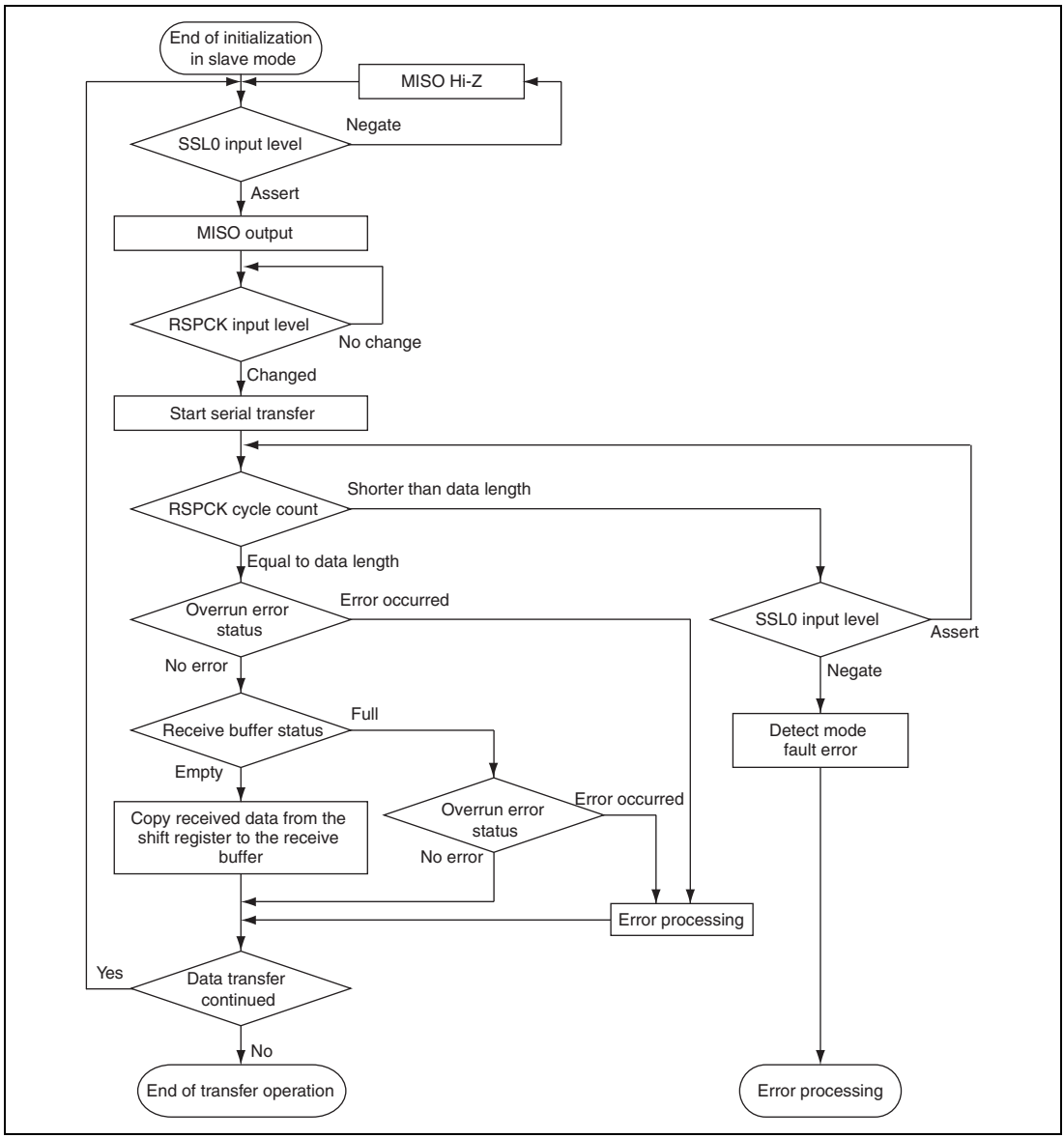


Figure 18.20 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

(2) Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 18.4.7, Error Detection). When operating in single-master mode (RSPI), 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-/multi-master modes.

(2-1) Starting Serial Transfer

The RSPI updates the data in the transmit buffer when the SPTEF bit in the RSPI status register (SPSR) is 1 and when either the CPU or the DTC/DMAC has written data to the RSPI data register (SPDR). If the shift register is empty in a condition where the SPTEF bit has been cleared to 0 due to the writing of 0 either after the writing to SPDR from the DTC/DMAC or by the writing of 0 after the value 1 is read from the SPTEF bit by the CPU, 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 18.4.4, Transfer Format. The polarity of the SSL output signal depends on the setting in the RSPI slave select polarity register (SSLP).

(2-2) 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 the SPRF bit in the RSPI status register (SPSR) is 0 and 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 settings in bits SPB3 to SPB0 in SPCMD. The polarity of the SSL output signal depends on the setting in the RSPI slave select polarity register (SSLP). For details on the RSPI transfer format, see section 18.4.4, Transfer Format.

(2-3) Sequence Control

The transfer format that is employed in master mode is determined by the RSPI sequence control register (SPSCR), RSPI command registers 0 to 3 (SPCMD0 to SPCMD3), the RSPI bit rate register (SPBR), the RSPI clock delay register (SPCKD), the RSPI slave select negation delay register (SSLND), and the RSPI next-access delay register (SPND).

The SPSCR register is used to determine the sequence configuration for serial transfers that are executed by a master mode RSPI. The following items are set in RSPI command registers SPCMD0 to SPCMD3: 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 SPCMD3. The RSPI contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP[1:0] in the RSPI sequence status register (SPSSR). 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.

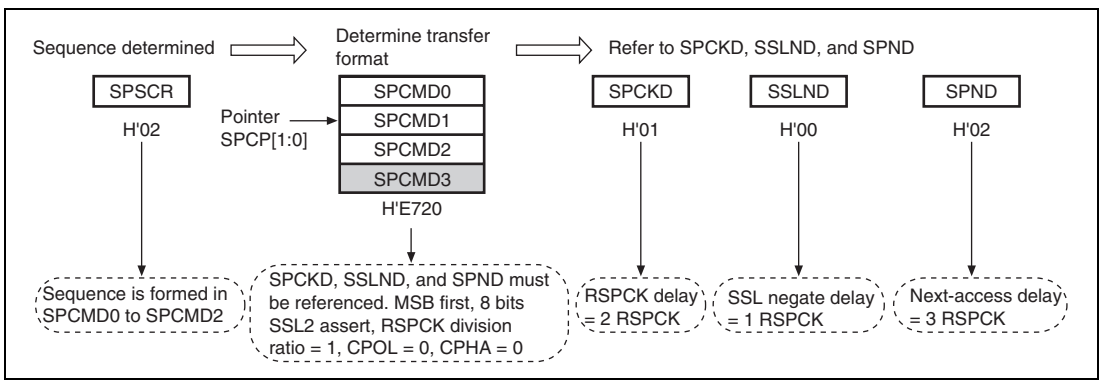


Figure 18.21 Determination Procedure of Serial Transfer Mode in Master Mode

(2-4) 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 SSL signal assertion status (burst transfer).

Figure 18.22 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 depicted in figure 18.22. 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 for next-access delay of SPCMD0 + 2 Pϕ at a minimum. If the shift register is empty after the passage of a minimum period, this period is sustained until such time as the transmit data is stored in the shift register for another transfer.
5. Based on SPCMD1, the RSPI asserts the SSL 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.

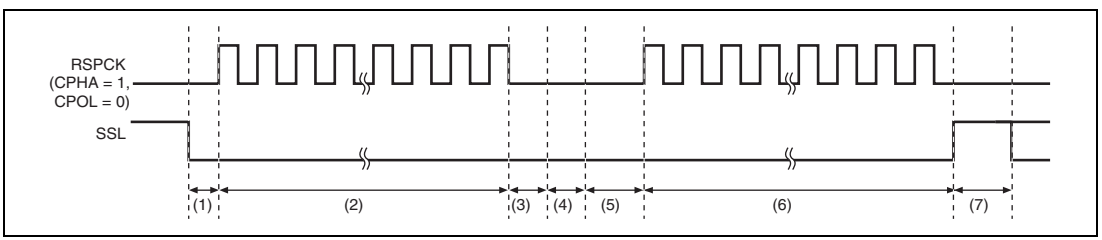


Figure 18.22 Example of Burst Transfer Operation using SSLKP Bit

If the SSL signal 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 18.22) corresponding to the command for the next transfer. Notice that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and the possibility arises of the collision of signal levels.

The RSPI in master mode references within the module the SSL signal operation for the case where the SSLKP bit is not used. Even when the CPHA bit in SPCMD is 0, the RSPI can accurately start serial transfers by asserting the SSL signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 18.4.9, SPI Operation).

(2-5) RSPCK Delay (t1)

The RSPCK delay value of the RSPI in master mode depends on SCKDEN bit settings in the RSPI command register (SPCMD) and on RSPCK delay register (SPCKD) settings. 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 18.9. For a definition of RSPCK delay, see section 18.4.4, Transfer Format.

Table 18.9 Relationship among SCKDEN and SPCKD Settings and RSPCK Delay Values

SCKDEN	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

(2-6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPI in master mode depends on SLNDEN bit settings in the RSPI command register (SPCMD) and on SSL negation delay register (SSLND) settings. 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 18.10. For a definition of SSL negation delay, see section 18.4.4, Transfer Format.

Table 18.10 Relationship among SLNDEN and SSLND Settings and SSL Negation Delay Values

SLNDEN	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

(2-7) Next-Access Delay (t3)

The next-access delay value of the RSPI in master mode depends on SPNDEN bit settings in the RSPI command register (SPCMD) and on next-access delay register (SPND) settings. 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 18.11. For a definition of next-access delay, see section 18.4.4, Transfer Format.

Table 18.11 Relationship among SPNDEN and SPND Settings and Next-Access Delay Values

SPNDEN	SPND	Next-Access 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

(2-8) Initialization Flowchart

Figure 18.23 shows an example of initialization flowchart for using the RSPI in master mode during SPI operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

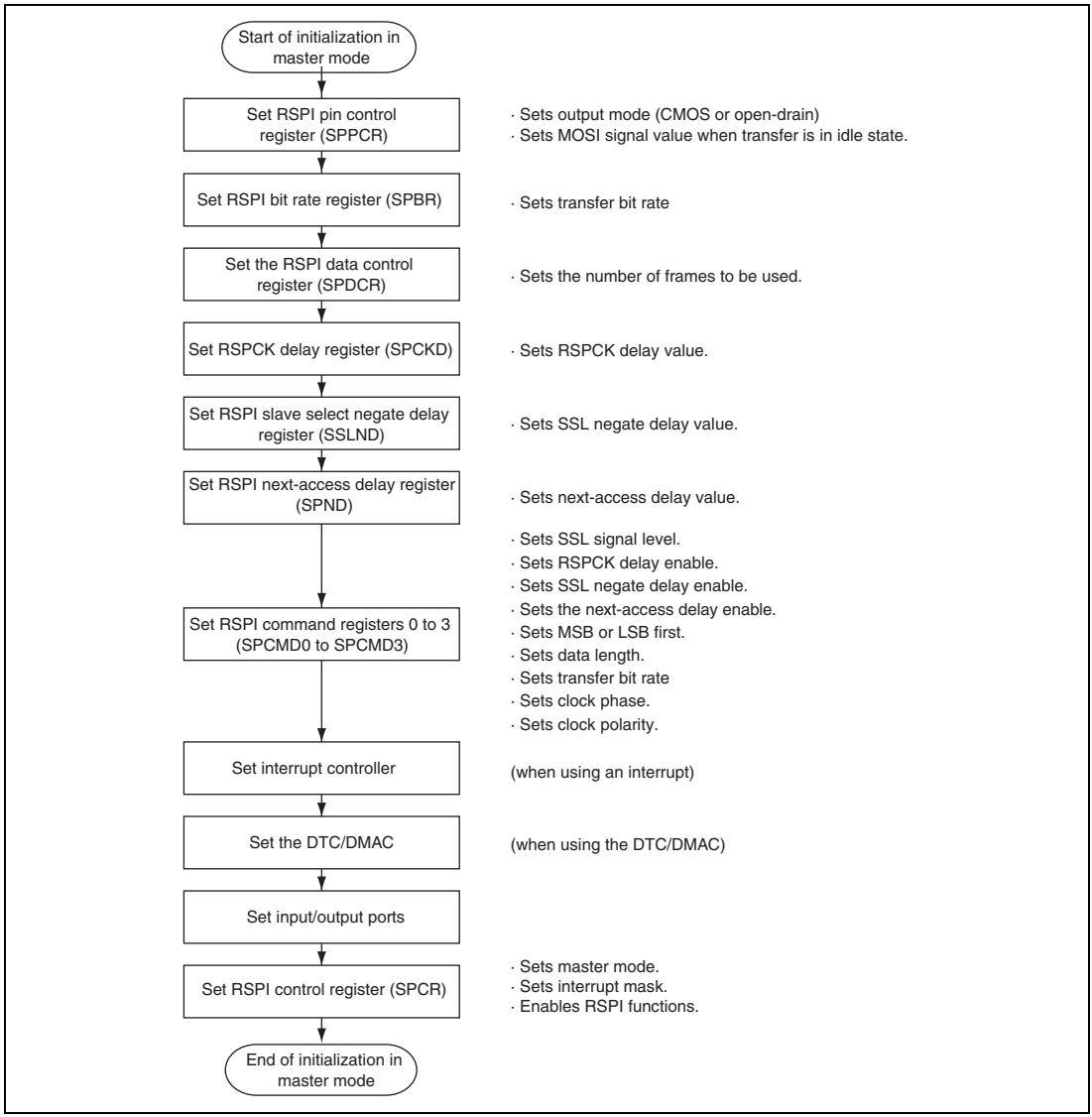
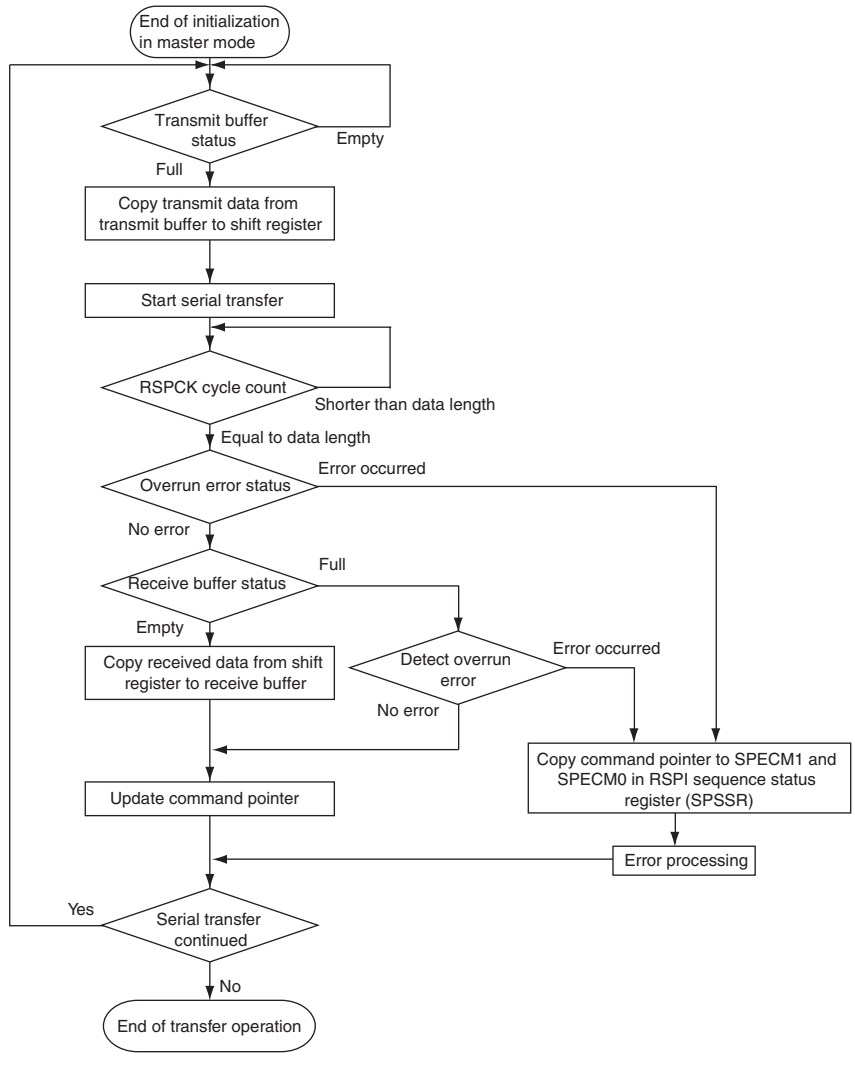


Figure 18.23 Example of Initialization Flowchart in Master Mode

(2-9) Transfer Operation Flowchart

Figure 18.24 shows an example of transfer operation flowchart for using the RSPI in master mode during SPI operation.



Note: When the SSL0 input signal is asserted in multi-master mode, the RSPI detects a mode fault error irrespective of the hardware status. When detecting the error, the RSPI copies the command pointer to the SPECM[1:0] bits in the RSPI sequence status register (SPSSR).

Figure 18.24 Example of Transfer Operation Flowchart in Master Mode

18.4.10 Clock Synchronous Operation

The RSPI selects clock synchronous operation when the SPMS bit in the RSPI control register (SPCR) is 1. During clock synchronous operation, the SSL pins are not used and the remaining three pins, RSPCK, MOSI, and MISO are used for communication. The SSL pins can be used as IO ports.

Although the SSL pins are not used for communication in clock synchronous operation, the internal operations within the modules are the same as those during SPI operation.

In both master and slave modes, communications can be performed with the same flows as the SPI operation except that mode fault error detection is not supported because the SSL pins are not used.

If the CPHA bit in the RSPI command register (SPCMD) is set in clock synchronous mode, operation cannot be guaranteed.

(1) Slave Mode Operation

(1-1) Starting a Serial Transfer

When the SPMS bit in the RSPI control register (SPCR) is 1, the first RSPCK edge 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 intact, 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 18.4.4, Transfer Format. Note that the SSL0 input signal is not used in clock synchronous operation.

(1-2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the SPRF bit in the RSPI status register (SPSR) is 0 and 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). Irrespective of the value of the SPRF bit, 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 the transfer data. In slave mode, the RSPI data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. For details on the RSPI transfer format, see section 18.4.4, Transfer Format.

(1-3) Initialization Flowchart

Figure 18.25 shows an example of initialization flowchart for using the RSPI in slave mode during clock synchronous operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

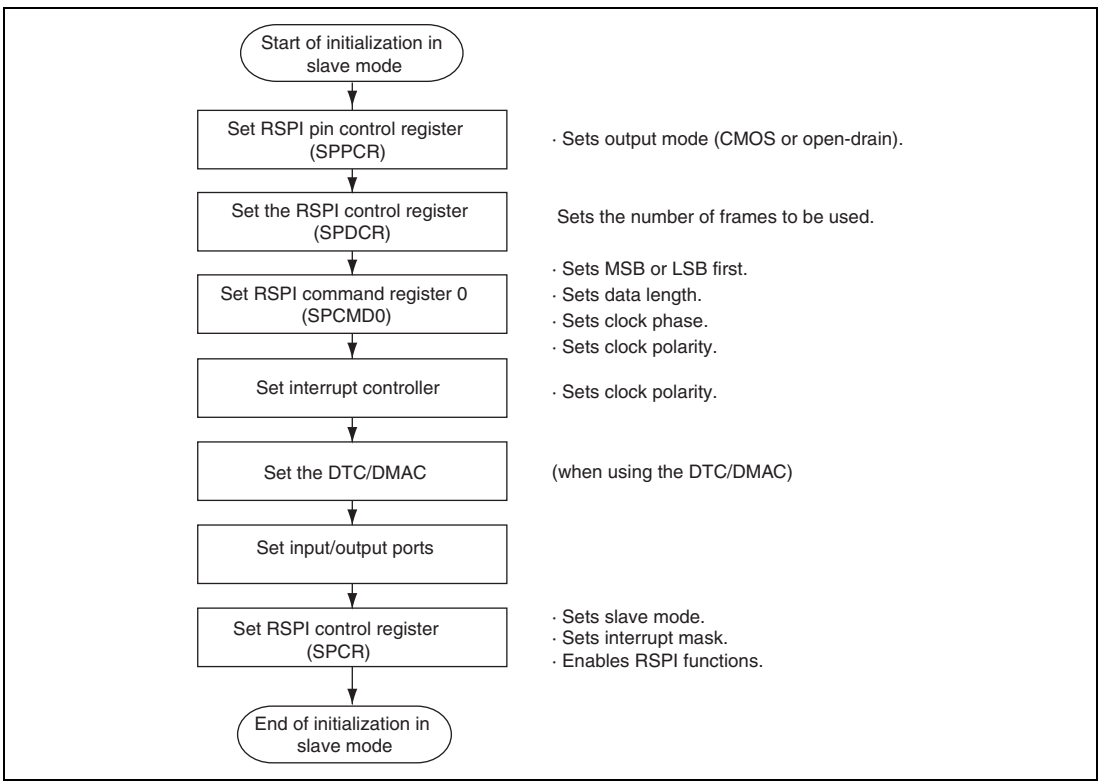


Figure 18.25 Example of Initialization Flowchart in Slave Mode

(1-4) Transfer Operation Flowchart (CPHA = 1)

Figure 18.26 shows an example of transfer operation flowchart for the RSPI during clock synchronous operation.

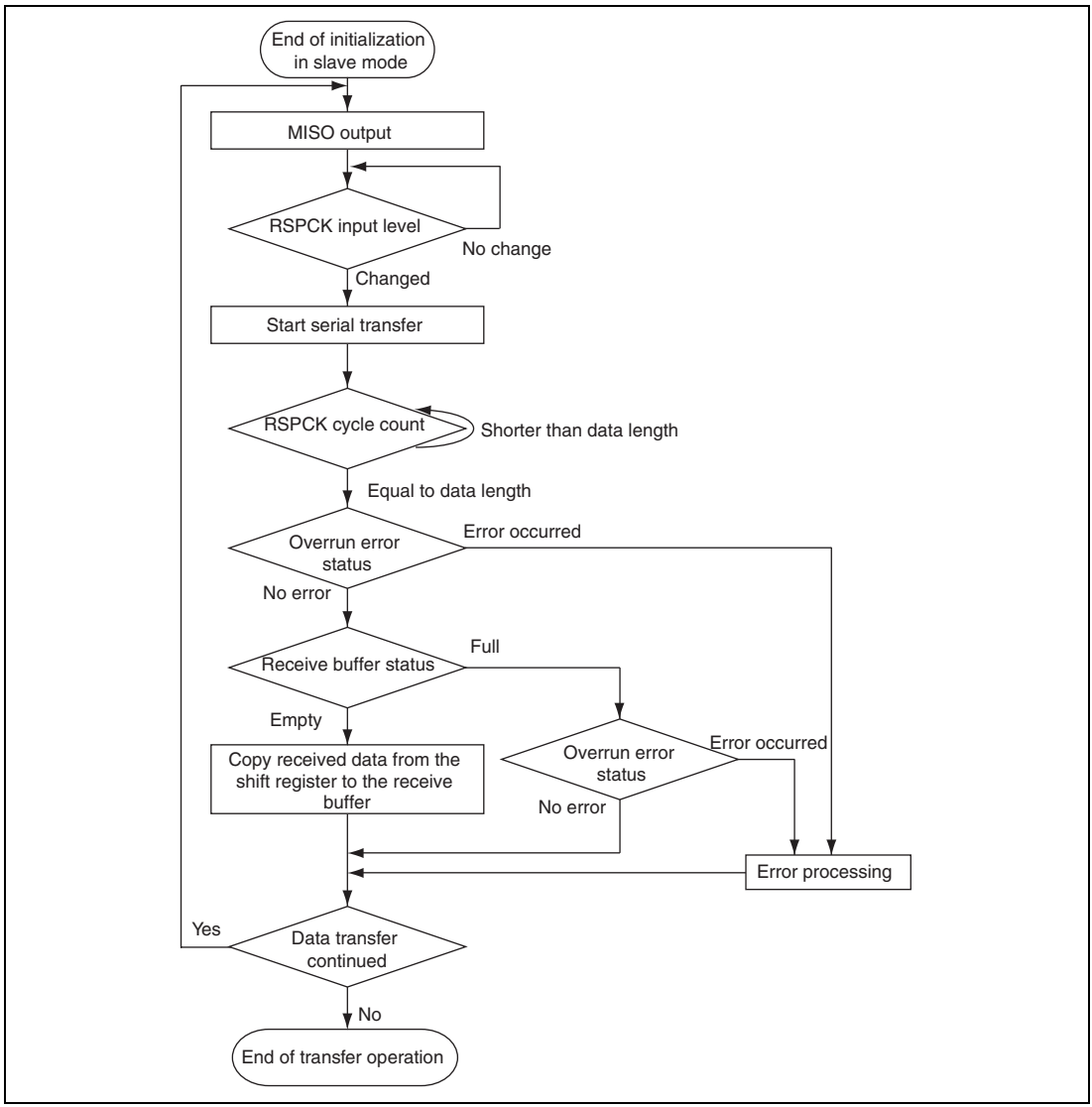


Figure 18.26 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

(2) Master Mode Operation

(2-1) Starting Serial Transfer

The RSPI updates the data in the transmit buffer when the SPTEF bit in the RSPI status register (SPSR) is 1 and when either the CPU or the DTC/DMAC has written data to the RSPI data register (SPDR). If the shift register is empty in a condition where the SPTEF bit has been cleared to 0 due to the writing of 0 either after the writing to SPDR from the DTC/DMAC or by the writing of 0 after the value 1 is read from the SPTEF bit by the CPU, 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 18.4.4, Transfer Format. Note that the SSL0 output signal is not used for communication in clock synchronous operation.

(2-2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. If the SPRF bit in the RSPI status register (SPSR) is 0 and 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 settings in bits SPB3 to SPB0 in SPCMD. For details on the RSPI transfer format, see section 18.4.4, Transfer Format. Note that the SSL0 output signal is not used for communication in clock synchronous operation.

(2-3) Sequence Control

The transfer format that is employed in master mode is determined by the RSPI sequence control register (SPSCR), RSPI command registers 0 to 3 (SPCMD0 to SPCMD3), the RSPI bit rate register (SPBR), the RSPI clock delay register (SPCKD), the RSPI slave select negation delay register (SSLND), and the RSPI next-access delay register (SPND). Although no SSL signal is output in clock synchronous operation, these settings are valid.

The SPSCR register is used to determine the sequence configuration for serial transfers that are executed by a master mode RSPI. The following items are set in RSPI command registers SPCMD0 to SPCMD3: 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 SPCMD3. The RSPI contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP[1:0] in the RSPI sequence status register (SPSSR). 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.

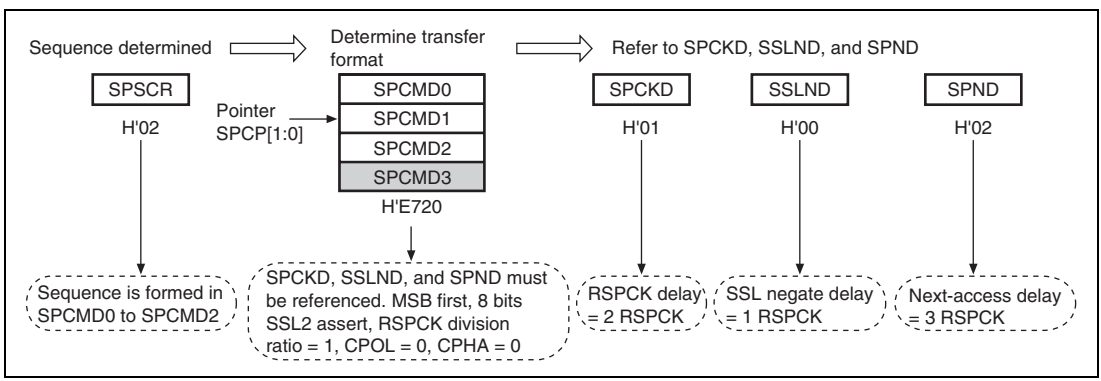


Figure 18.27 Determination Procedure of Serial Transfer Mode in Master Mode

(2-4) Initialization Flowchart

Figure 18.28 shows an example of initialization flowchart for using the RSPI in master mode during clock synchronous operation. For a description of how to set up an interrupt controller, the DTC/DMAC, and input/output ports, see the descriptions given in the individual blocks.

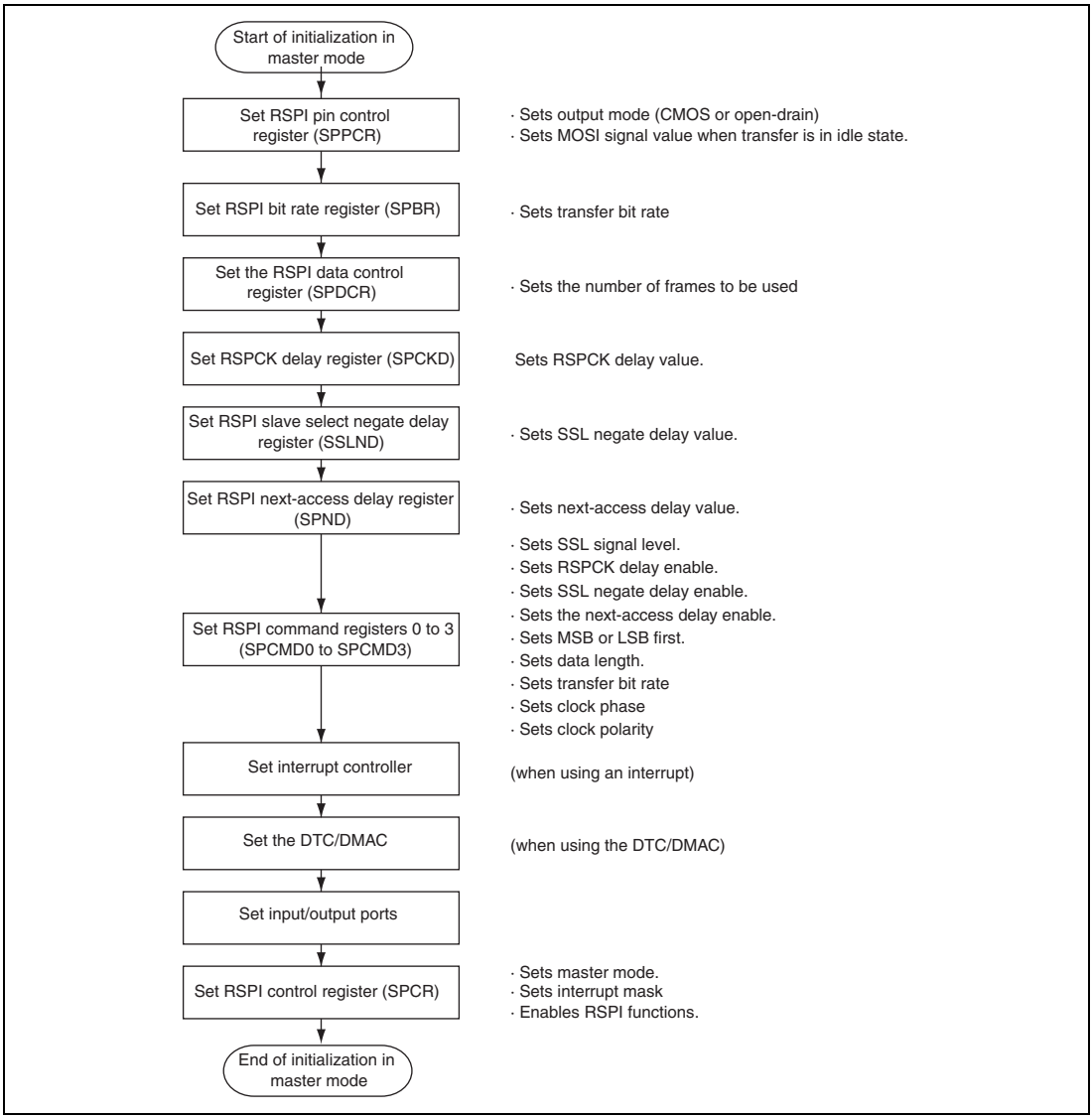


Figure 18.28 Example of Initialization Flowchart in Master Mode

(2-5) Transfer Operation Flowchart

Figure 18.29 shows an example of transfer operation flowchart in master mode during clock synchronous operation.

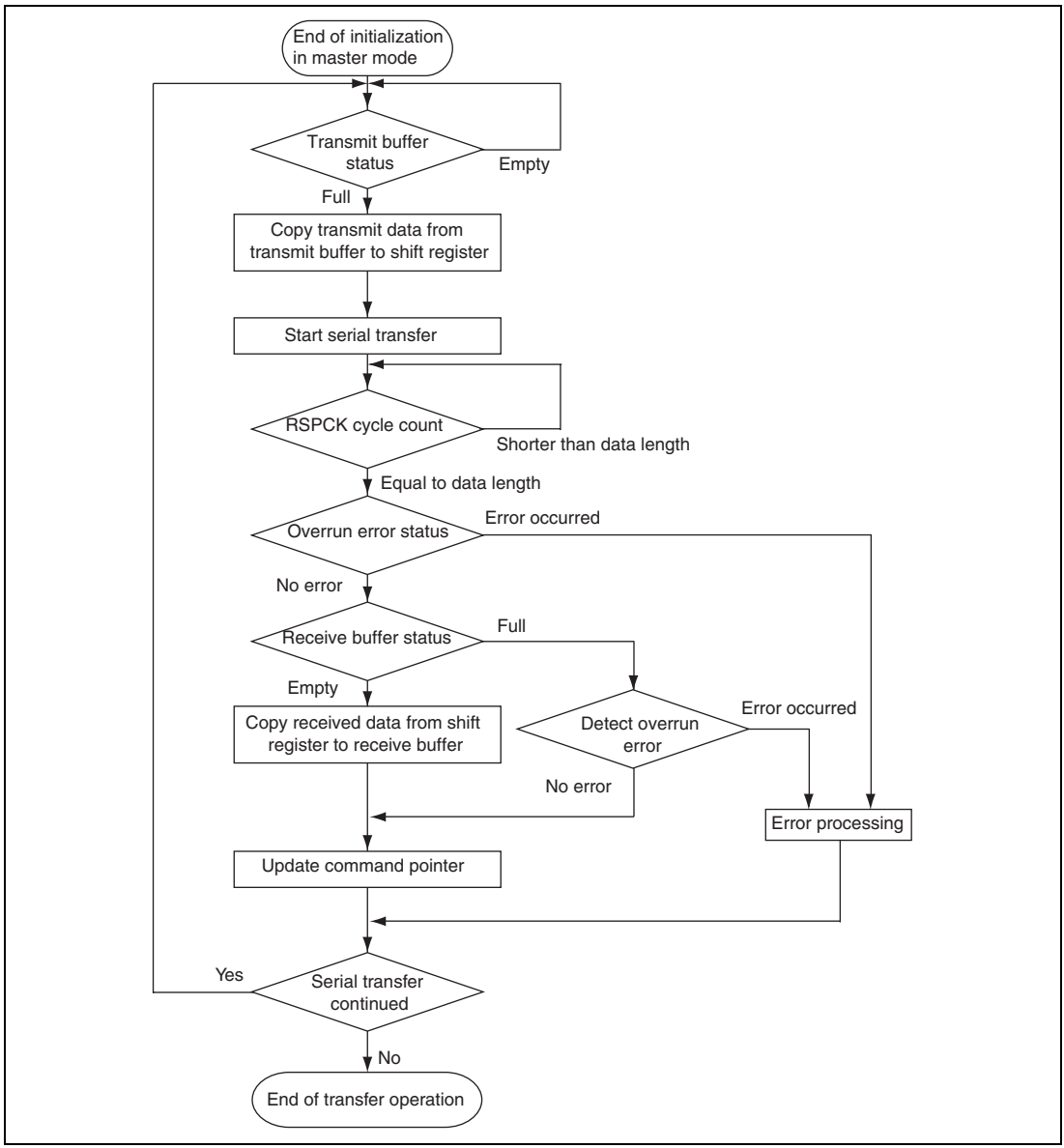


Figure 18.29 Example of Transfer Operation Flowchart in Master Mode

18.4.11 Error Processing

Figures 18.30 and 18.31 show error processing. The RSPI can recover from an error which may occur in master or slave mode, using the following error processing.

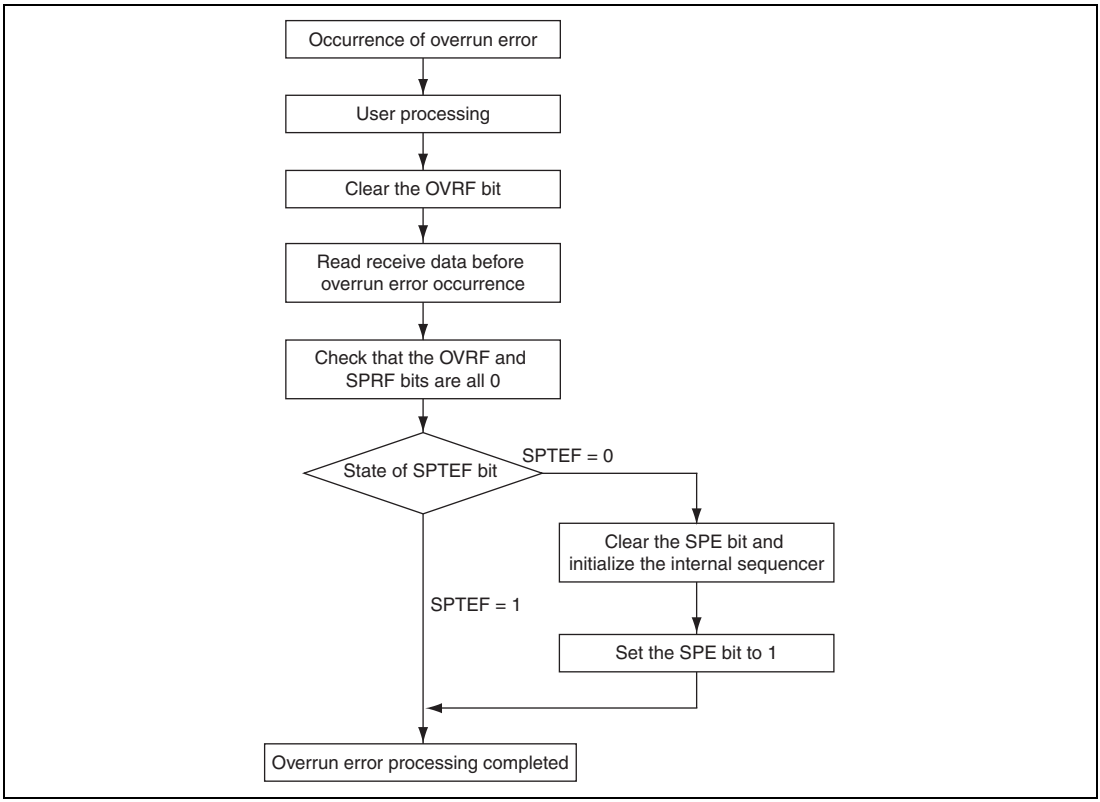


Figure 18.30 Error Processing (Overrun Error)

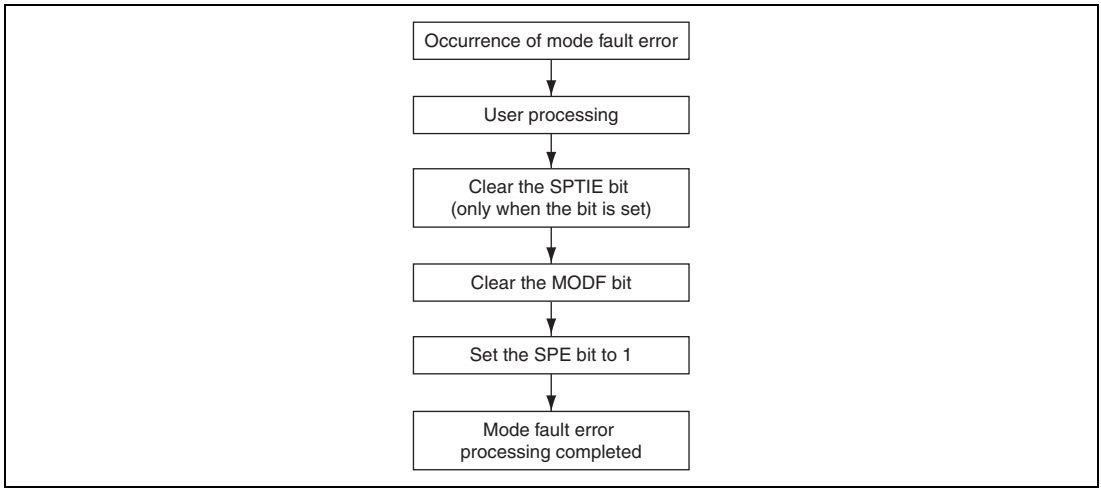


Figure 18.31 Error Processing (Mode Fault Error)

18.4.12 Loopback Mode

When the CPU writes 1 to the SPLP bit in the RSPI pin control register (SPPCR), the RSPI shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects the input path and the output path (reversed) of the shift register. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI becomes the received data for the RSPI. Figure 18.32 shows the configuration of the shift register input/output paths for the case where the RSPI in master mode is set in loopback mode.

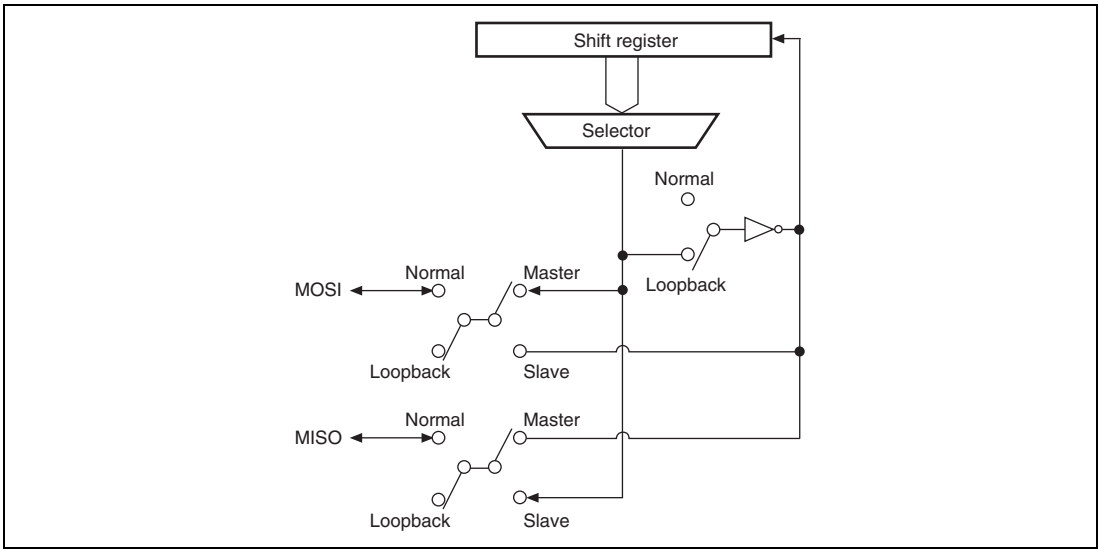


Figure 18.32 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

18.4.13 Interrupt Request

The interrupt sources for the RSPI include receive-buffer-full, transmission-buffer-empty, mode-fault, and overrun. With an interrupt request of receive-buffer-full or transmission-buffer-empty, the DTC or DMAC can start up and perform a data transfer.

The interrupt request of receive-buffer-full is allocated to the vector address of SPRI, the interrupt request of transmission-buffer-empty is allocated to the vector address of SPTI, and the interrupt requests of mode-fault and overrun are allocated to the vector address of SPEI. Therefore it is necessary to determine the interrupt source by the flag. Table 18.12 shows the interrupt sources for the RSPI.

When the interrupt condition is satisfied as shown in table 18.12, an interrupt occurs. Clear the interrupt source by executing a data transfer by the CPU or DTC/DMAC.

Table 18.12 RSPI Interrupt Sources

Name	Interrupt Source	Symbol	Interrupt Condition	DTC/DMAC Startup
SPRI	Receive-buffer-full	RXI	(SPRIE=1) • (SPRF=1)	Startup
SPTI	Transmission-buffer-empty	TXI	(SPTIE=1) • (SPTEF=1)	Startup
SPEI	Mode-fault	MOI	(SPEIE=1) • (MODF=1)	—
	Overrun	OVI	(SPEIE=1) • (OVRF=1)	—

18.5 Usage Notes

18.5.1 DTC Block Transfer

To start a DTC block transfer due to RXI and TXI, set the block size in the DTC transfer count register (CRA) and the value in the block size counter to the same value as the number of frames set in the frame count setting bit. If these values are not the same, subsequent operations cannot be guaranteed.

18.5.2 DMAC Burst Transfer

To start a DMAC transfer due to RXI and TXI, set the value in the DMA transfer count register (DMATCR) to the same value as the number of frames set in the frame count setting bit. If these values are not the same, subsequent operations cannot be guaranteed.

18.5.3 Reading Receive Data

When reading the receive data by the CPU, clear the flag after the CPU reads the buffer for the specified number of times. If the flag is cleared before reaching the specified number of times, subsequent operations cannot be guaranteed.

18.5.4 DTC/DMAC and Mode Fault Error

If a mode fault error occurs when the SPTXI interrupt setting for DTC/DMAC is enabled while the SPTIE bit is valid, an unintended interrupt may occur. Clear the SPTIE bit while it is valid using the mode fault error processing (figure 18.31).

To use the DTC/DMAC after a mode fault error occurrence, reset the DTC/DMAC.

18.5.5 Usage of the RSPI Output Pins as Open Drain Outputs

When the RSPI output pins are to be used as open drain outputs, use a pull-up register to pull them up to the same electric potential as that on the $V_{CC}Q$ pin.

Specify the pull-up resistance after enough evaluation to considerate whether the load satisfies the electrical characteristic requirements.

Section 19 I^2C Bus Interface 3 (IIC3)

The I^2C bus interface 3 conforms to and provides a subset of the Philips I^2C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I^2C bus differs partly from the Philips register configuration.

19.1 Features

- Selection of I^2C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.

I^2C bus format:

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- The direct memory access controller (DMAC) or data transfer controller (DTC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Clocked synchronous serial format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error
- The direct memory access controller (DMAC)) or data transfer controller (DTC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

Figure 19.1 shows a block diagram of the I²C bus interface 3.

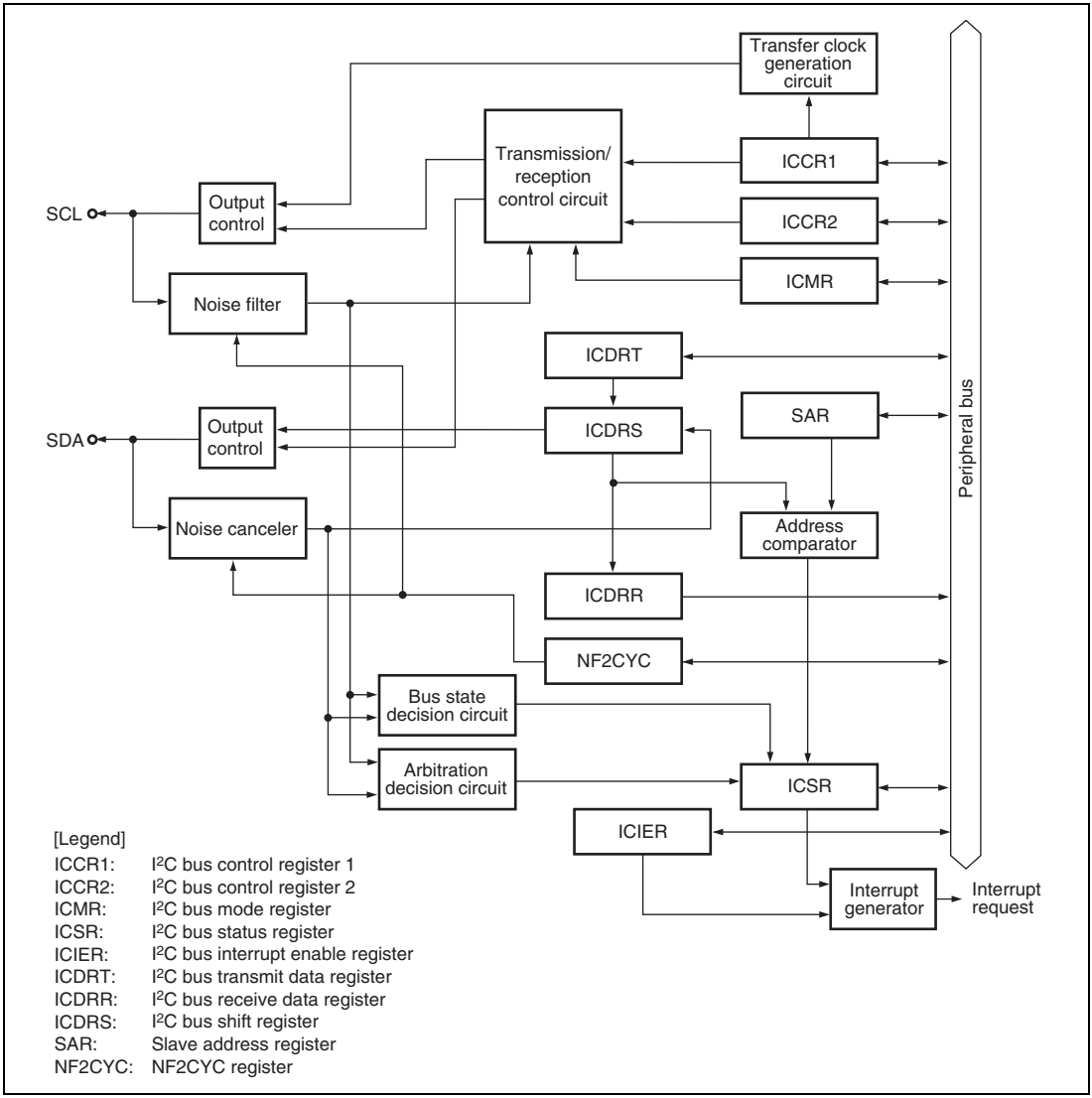


Figure 19.1 Block Diagram of I²C Bus Interface 3

19.2 Input/Output Pins

Table 19.1 shows the pin configuration of the I^2C bus interface 3.

Table 19.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock	SCL	I/O	I^2C serial clock input/output
Serial data	SDA	I/O	I^2C serial data input/output

Figure 19.2 shows an example of I/O pin connections to external circuits.

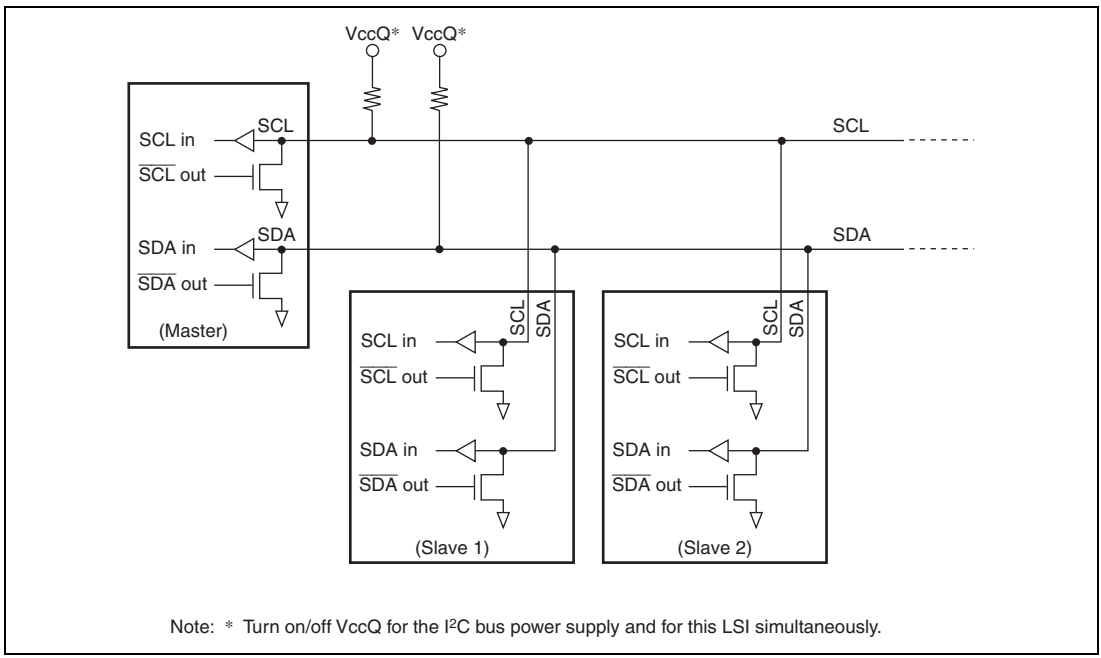


Figure 19.2 External Circuit Connections of I/O Pins

19.3 Register Descriptions

The I^{2}C bus interface 3 has the following registers.

Table 19.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
I^{2}C bus control register 1	ICCR1	R/W	H'00	H'FFFEE000	8
I^{2}C bus control register 2	ICCR2	R/W	H'7D	H'FFFEE001	8
I^{2}C bus mode register	ICMR	R/W	H'38	H'FFFEE002	8
I^{2}C bus interrupt enable register	ICIER	R/W	H'00	H'FFFEE003	8
I^{2}C bus status register	ICSR	R/W	H'00	H'FFFEE004	8
Slave address register	SAR	R/W	H'00	H'FFFEE005	8
I^{2}C bus transmit data register	ICDRT	R/W	H'FF	H'FFFEE006	8
I^{2}C bus receive data register	ICDRR	R/W	H'FF	H'FFFEE007	8
NF2CYC register	NF2CYC	R/W	H'00	H'FFFEE008	8

19.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 3, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

ICCR1 is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface 3 Enable 0: Output from SCL and SDA is disabled. (Input to SCL and SDA enabled .) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when TRS is 0 and ICDRR is read. In master receive mode, when ICDRR cannot be read before the rising edge of the 8th clock of SCL, set RCVD to 1 so that data is received in byte units. In other modes, clear this bit to 0. If RCVD is set to 1 so that data is received in byte units, read ICDRR after the falling edge of the 9th clock. 0: Enables next reception 1: Disables next reception

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I^2C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the 8th bit is set to 1, TRS is automatically set to 1. If an overrun error occurs in master receive mode with the clocked synchronous serial format, MST is cleared and the mode changes to slave receive mode. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST = 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 19.3) in master mode.

Table 19.3 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate		
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 40 MHz (160/8)	Pϕ = 48 MHz (160/6)	Pϕ = 50 MHz (160/4)
0	0	0	0	Pϕ/64	625	750	781
0	0	0	1	Pϕ/72	556	667	694
0	0	1	0	Pϕ/84	476	571	595
0	0	1	1	Pϕ/92	435	521	543
0	1	0	0	Pϕ/100	400	480	500
0	1	0	1	Pϕ/108	370	444	463
0	1	1	0	Pϕ/120	333	400	417
0	1	1	1	Pϕ/124	322	387	403
1	0	0	0	Pϕ/256	156	188	195
1	0	0	1	Pϕ/288	139	167	174
1	0	1	0	Pϕ/336	119	143	149
1	0	1	1	Pϕ/368	109	130	136
1	1	0	0	Pϕ/400	100	120	125
1	1	0	1	Pϕ/432	92.6	111	116
1	1	1	0	Pϕ/480	83.3	100	104
1	1	1	1	Pϕ/496	80.6	96.7	101

Note: The settings should satisfy external specifications.

19.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus.

ICCR2 is initialized to H'7D by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	-	IICRST	-
Initial value:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clocked synchronous serial format, this bit is always read as 0. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	R/W	Start/Stop Issue Condition Disable Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. Even if 1 is written to this bit, the data will not be stored.

Bit	Bit Name	Initial Value	R/W	Description
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).
4	SDAOP	1	R/W	SDAO Write Protect Controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0. This bit is always read as 1.
3	SCLO	1	R	SCL Output Level Monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	IICRST	0	R/W	IIC Control Part Reset Resets the control part except for I^2C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I^2C bus operation, some IIC3 registers and the control part can be reset.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

19.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, and selects the transfer bit count.

ICMR is initialized to H'38 by a power-on reset. Bits BC[2:0] are initialized to H'0 by the IICRST bit in ICCR2.

Bit:	7	6	5	4	3	2	1	0
	MLS	-	-	-	BCWP	BC[2:0]		
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the BC[2:0] modifications. When modifying the BC[2:0] bits, this bit should be cleared to 0. In clocked synchronous serial mode, the BC[2:0] bits should not be modified. 0: When writing, values of the BC[2:0] bits are set. 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I^2C bus format, the data is transferred with one addition acknowledge bit. Should be made between transfer frames. If these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. After the stop condition is detected, the value of these bits returns automatically to B'111. The value returns to B'000 at the end of a data transfer, including the acknowledge bit. These bits are cleared by a power-on reset and in software standby mode and module standby mode. These bits are also cleared by setting the IICRST bit of ICCR2 to 1. With the clocked synchronous serial format, these bits should not be modified. I^2C Bus Format — Clocked Synchronous Serial Format 000: 9 bits — 000: 8 bits 001: 2 bits — 001: 1 bit 010: 3 bits — 010: 2 bits 011: 4 bits — 011: 3 bits 100: 5 bits — 100: 4 bits 101: 6 bits — 101: 5 bits 110: 7 bits — 110: 6 bits 111: 8 bits — 111: 7 bits

19.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits received.

ICIER is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection and arbitration lost/overrun error interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: Disables the NACK detection and arbitration lost/overrun error interrupt request (NAKI). 1: Enables the NACK detection and arbitration lost/overrun error interrupt request (NAKI).
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (STPI) when the STOP bit in ICSR is set. 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. This bit can be canceled by setting the BBSY bit in ICCR2 to 1. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

19.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

ICSR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clocked synchronous serial format

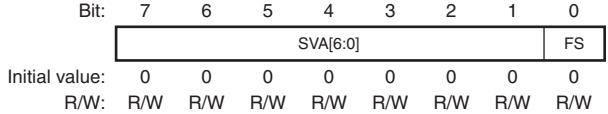
Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1 [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting conditions] • In master mode, when a stop condition is detected after frame transfer • In slave mode, when the slave address in the first byte after the general call and detecting start condition matches the address set in SAR, and then the stop condition is detected

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag Indicates that arbitration was lost in master mode with the I^2C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I^2C bus interface 3 detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master. [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1 [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected • When the final bit is received with the clocked synchronous format while RDRF = 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA[6:0] in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS = 1 [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode.
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode with the I^2C bus format. [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1 [Setting condition] • When the general call address is detected in slave receive mode

19.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.

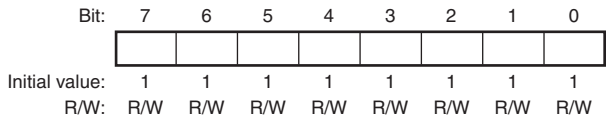
SAR is initialized to H'00 by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is selected

19.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. ICDRT is initialized to H'FF.



19.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.

ICDRR is initialized to H'FF by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

Bit:	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

19.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 19.4.7, Noise Filter.

NF2CYC is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	NF2CYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the peripheral clock can be filtered out 1: The noise less than two cycles of the peripheral clock can be filtered out

19.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

19.4.1 I²C Bus Format

Figure 19.3 shows the I²C bus formats. Figure 19.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

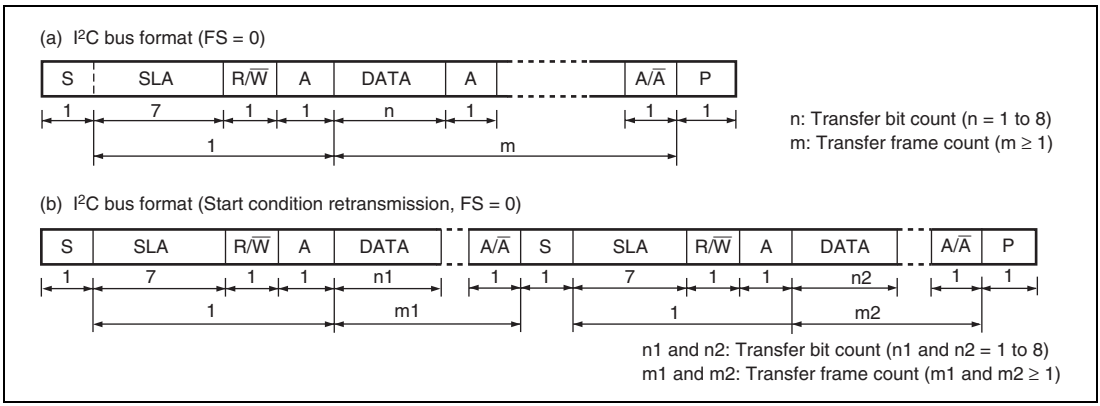


Figure 19.3 I²C Bus Formats

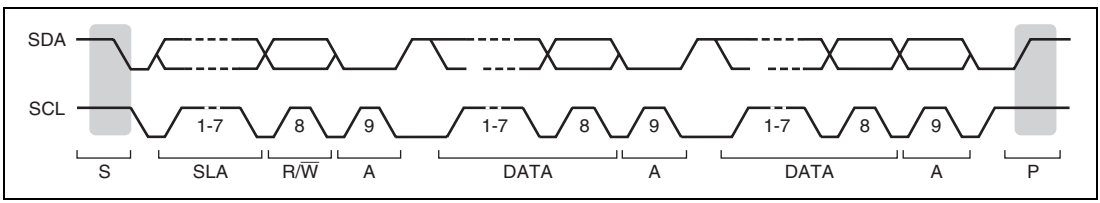


Figure 19.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

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 SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

19.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 19.5 and 19.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set ICMR and bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to slave receive mode.

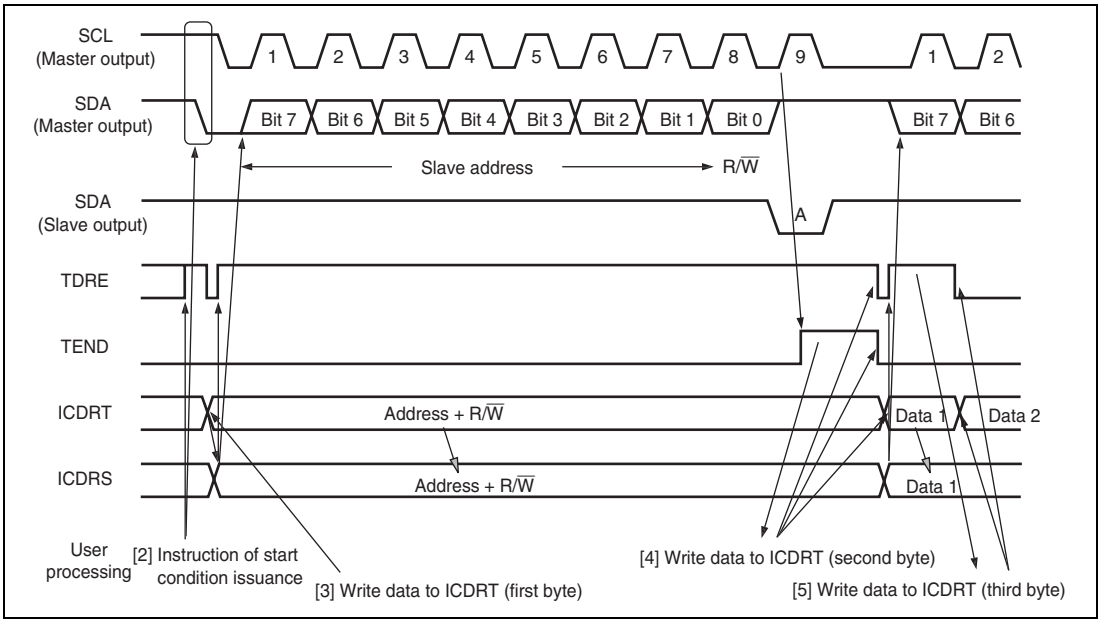


Figure 19.5 Master Transmit Mode Operation Timing (1)

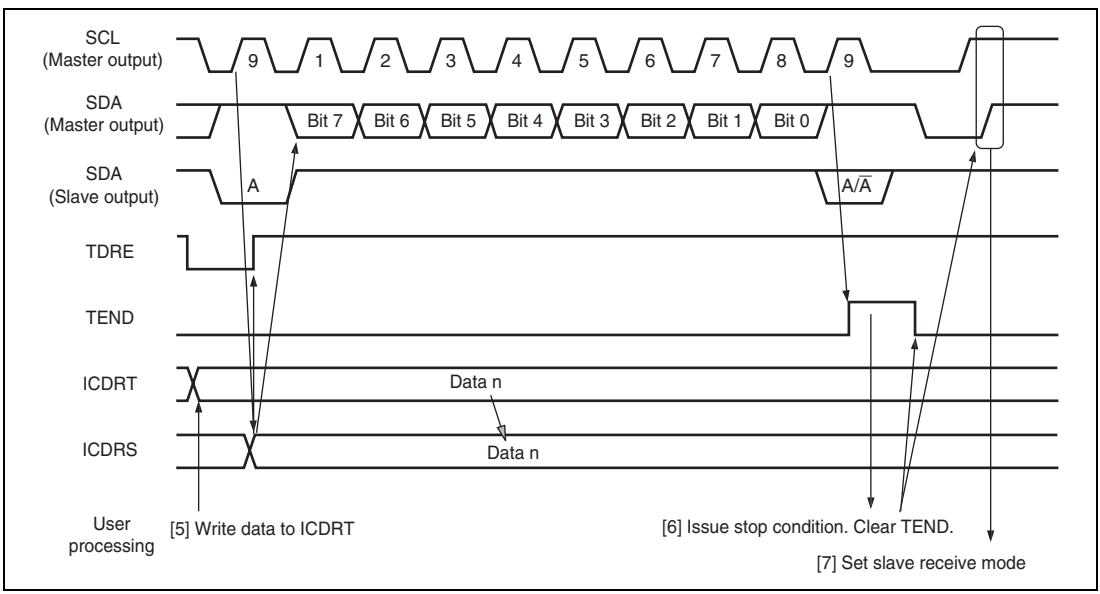


Figure 19.6 Master Transmit Mode Operation Timing (2)

19.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 19.7 and 19.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

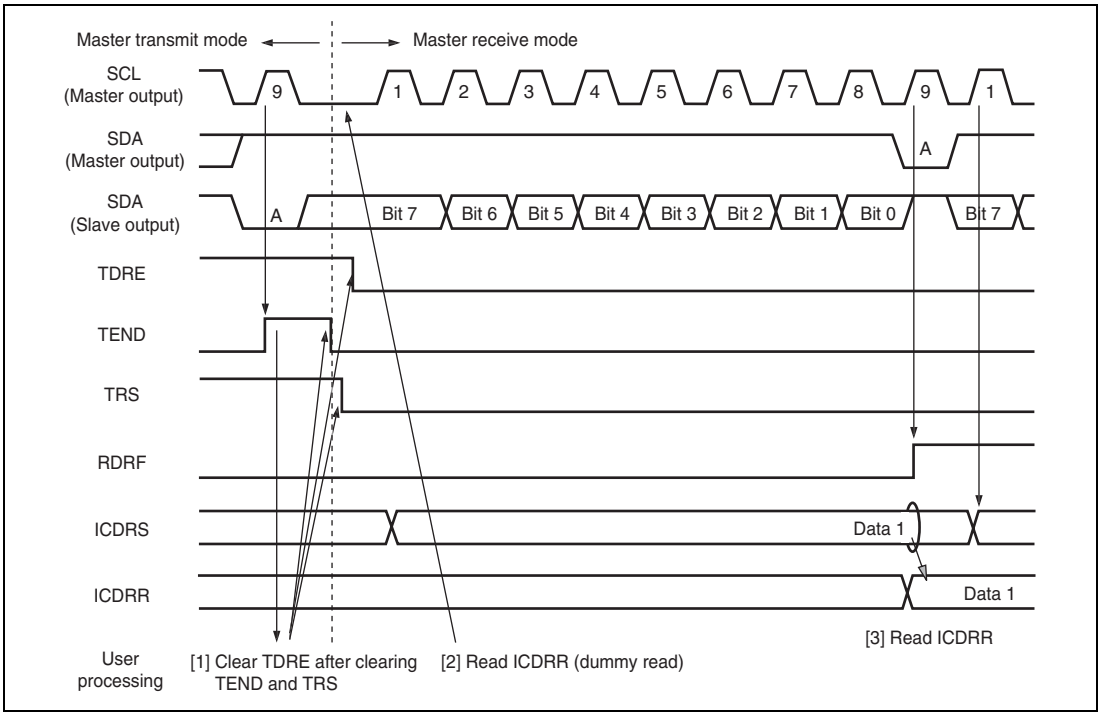


Figure 19.7 Master Receive Mode Operation Timing (1)

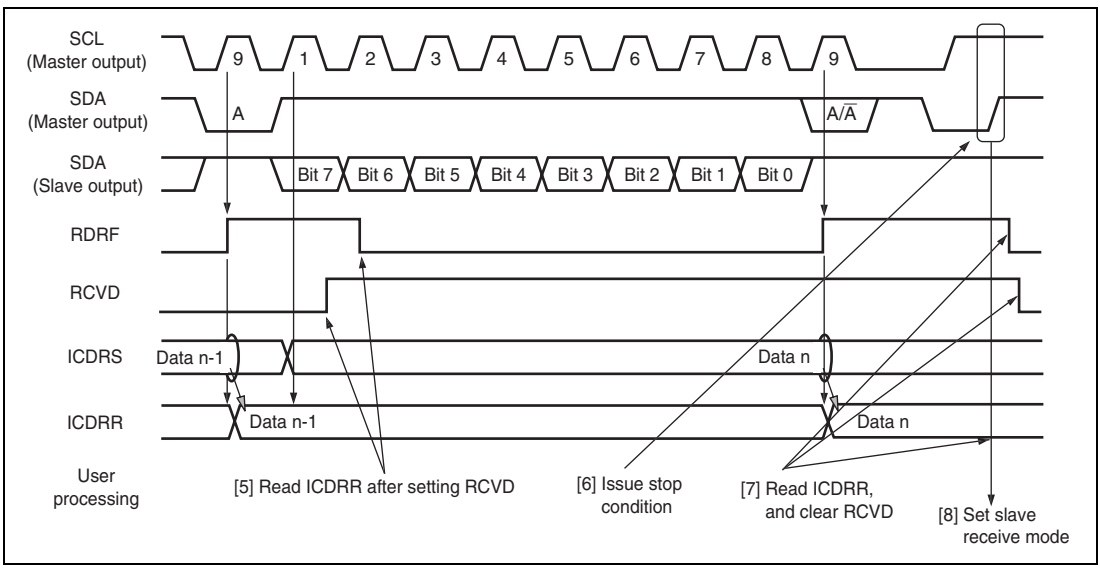


Figure 19.8 Master Receive Mode Operation Timing (2)

19.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 19.9 and 19.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS bit in ICCR1 and the TDRE bit in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is opened.
5. Clear TDRE.

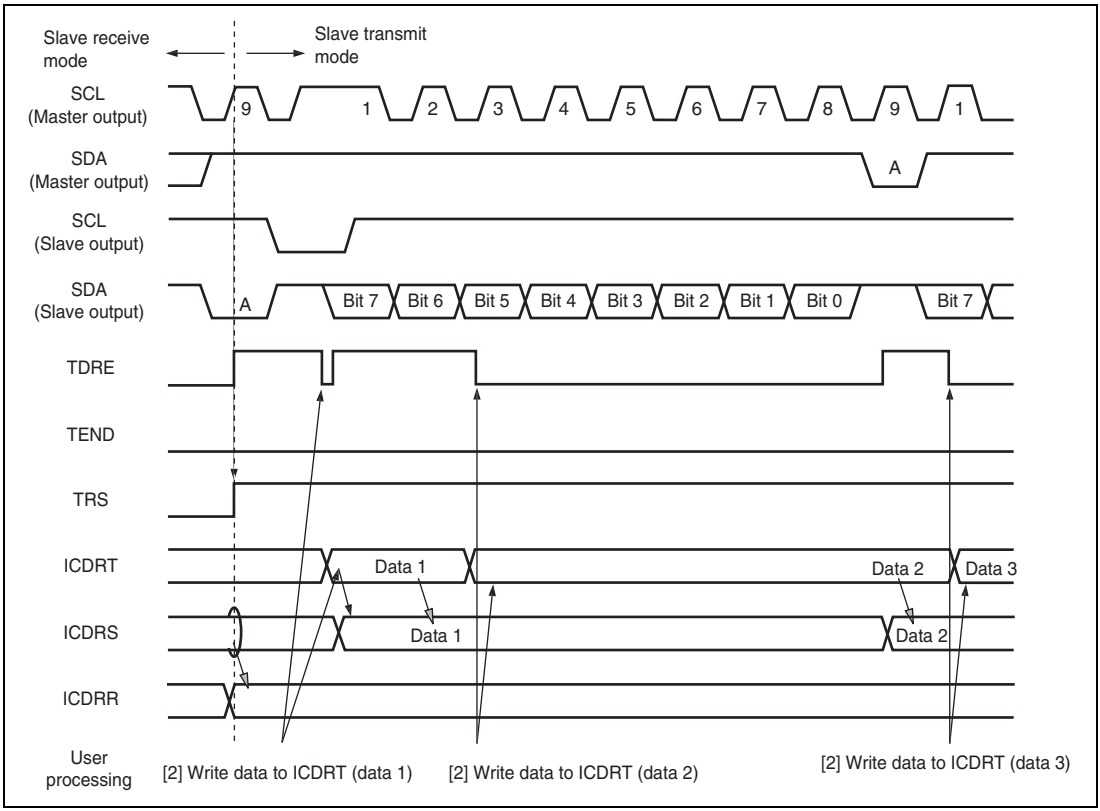


Figure 19.9 Slave Transmit Mode Operation Timing (1)

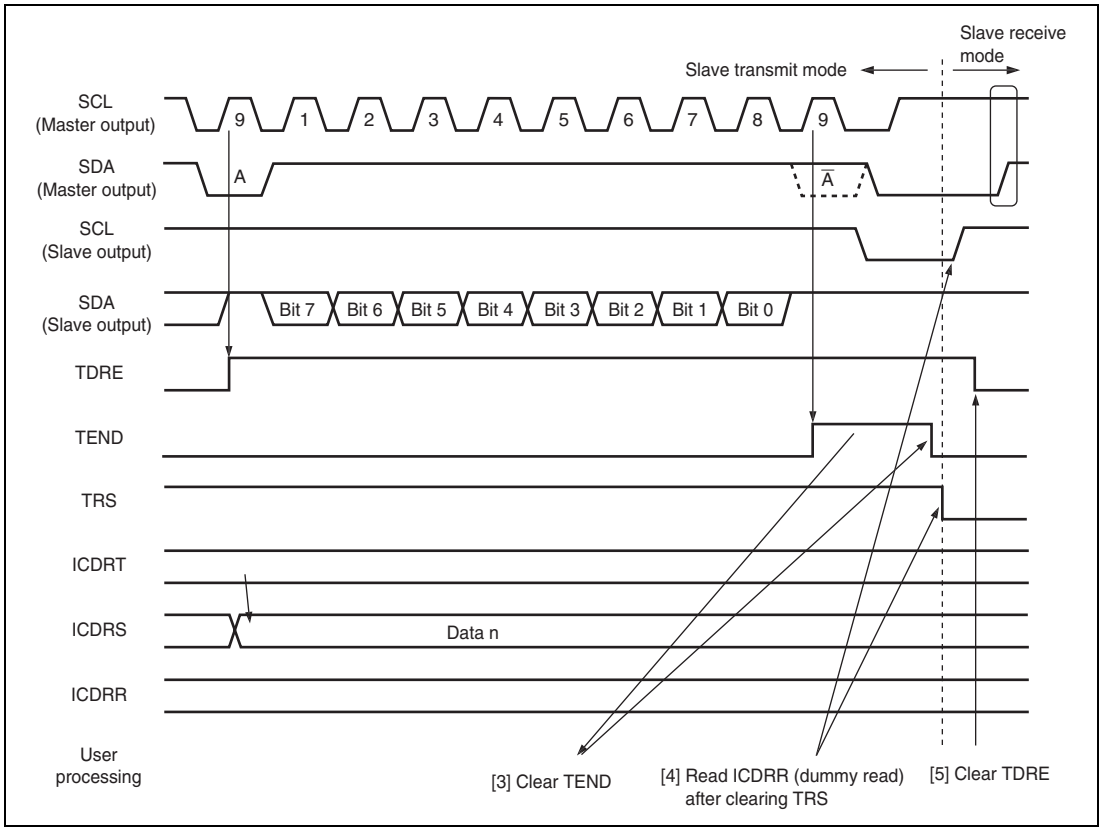


Figure 19.10 Slave Transmit Mode Operation Timing (2)

19.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 19.11 and 19.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

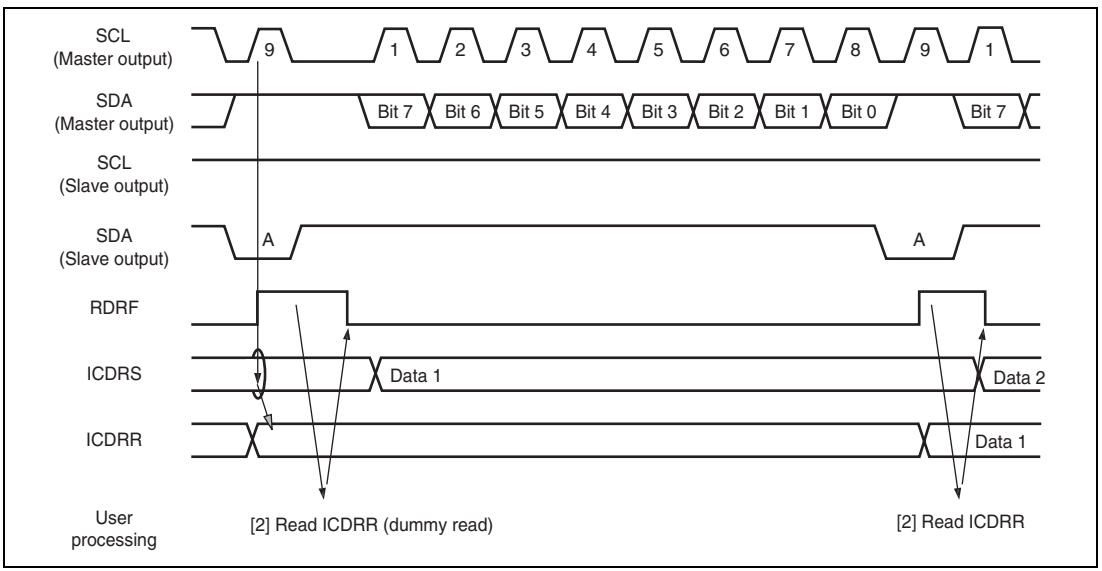


Figure 19.11 Slave Receive Mode Operation Timing (1)

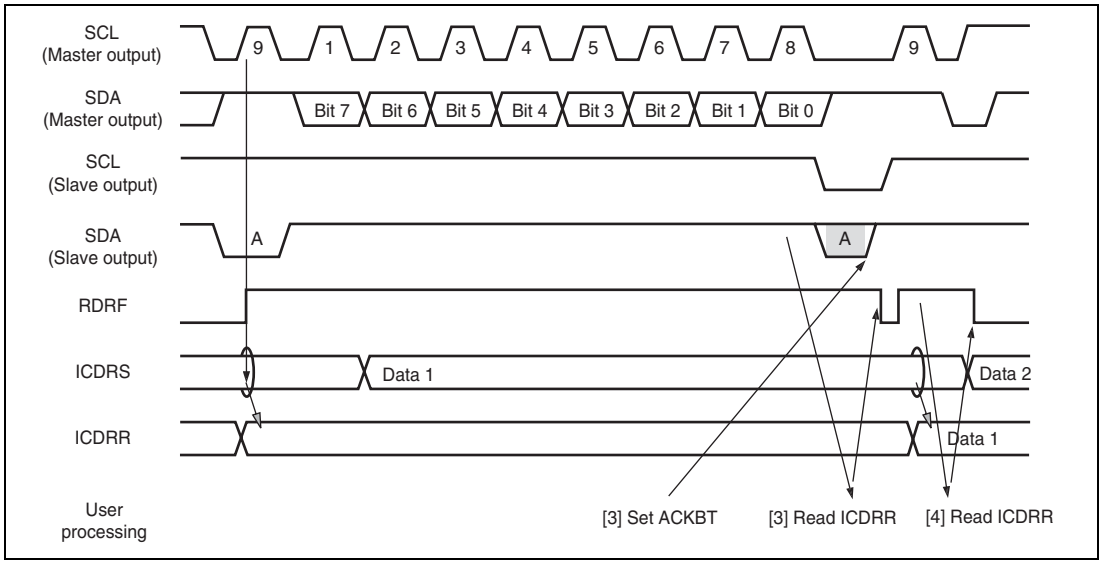


Figure 19.12 Slave Receive Mode Operation Timing (2)

19.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 19.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the fall to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

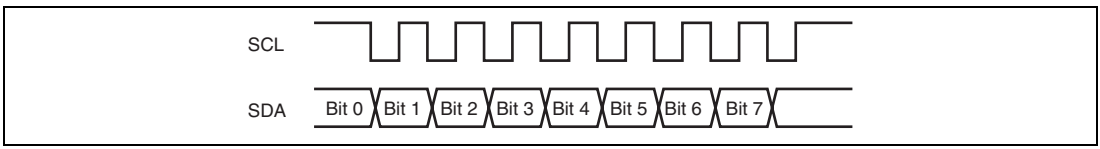


Figure 19.13 Clocked Synchronous Serial Transfer Format

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 19.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS[3:0] bits in ICCR1. (Initial setting)
2. Set the TRS bit in ICCR1 to select transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

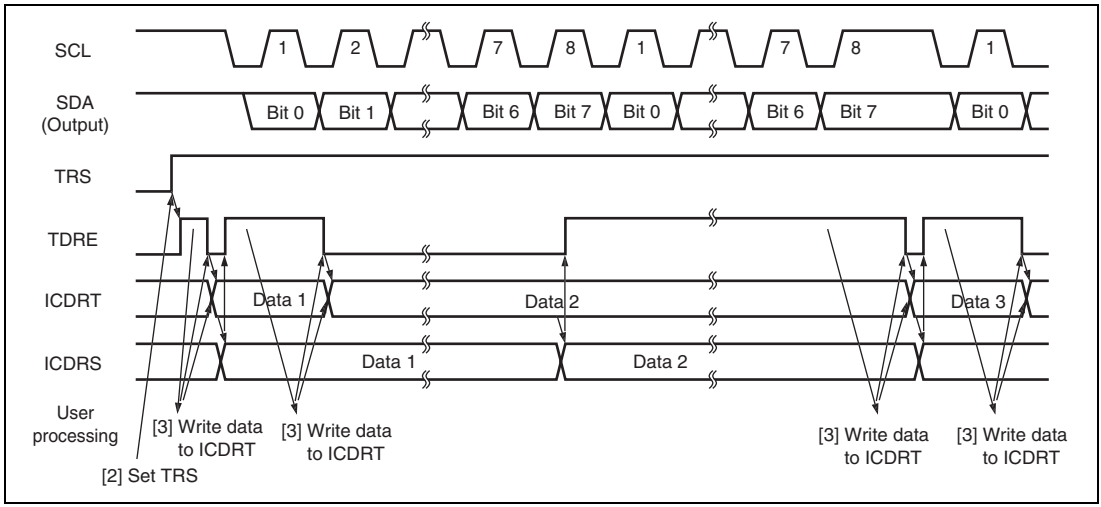


Figure 19.14 Transmit Mode Operation Timing

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 19.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock rises while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST = 1 specified. See figure 19.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. Set MST = 1 while the RCVD bit in ICCR1 is 0. This causes the receive clock to be output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 to 1. This causes the SCL to be fixed to the high level after outputting one byte of the receive clock.

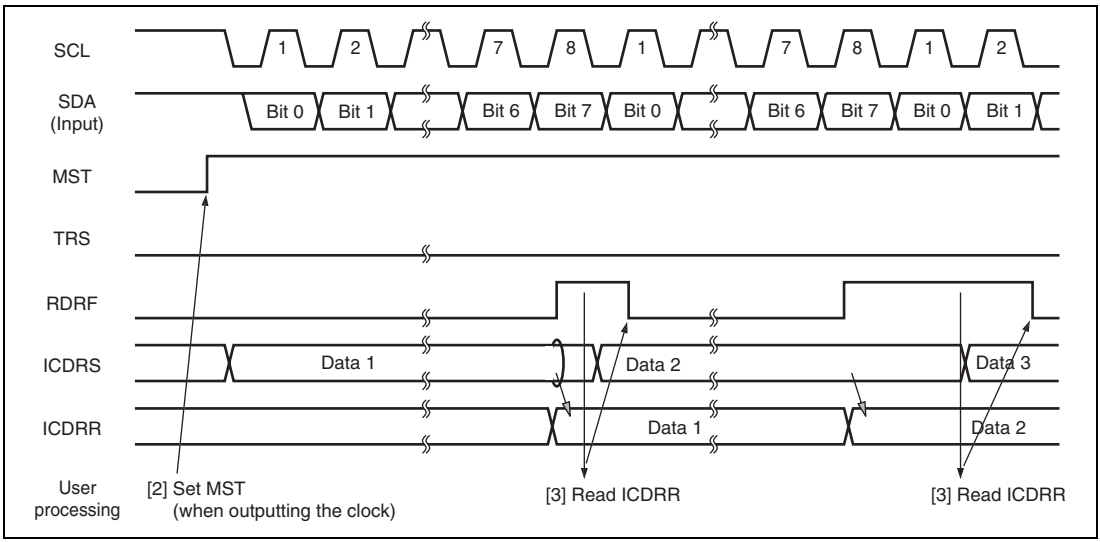


Figure 19.15 Receive Mode Operation Timing

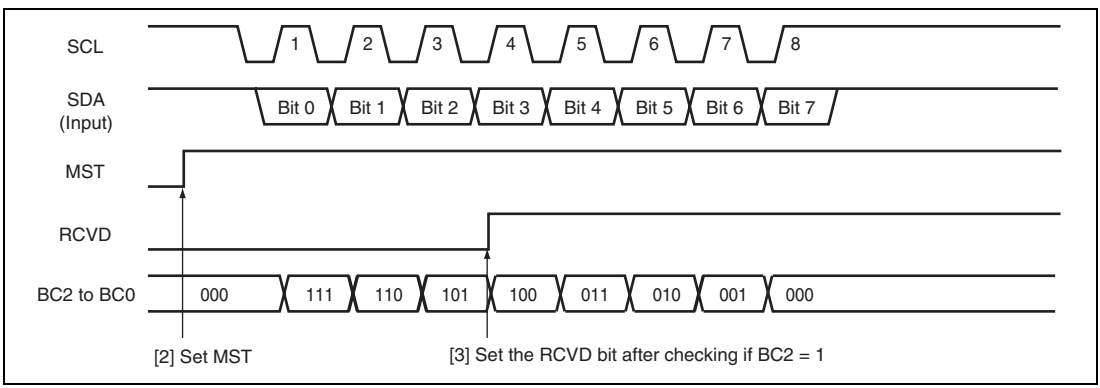


Figure 19.16 Operation Timing For Receiving One Byte (MST = 1)

19.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 19.17 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

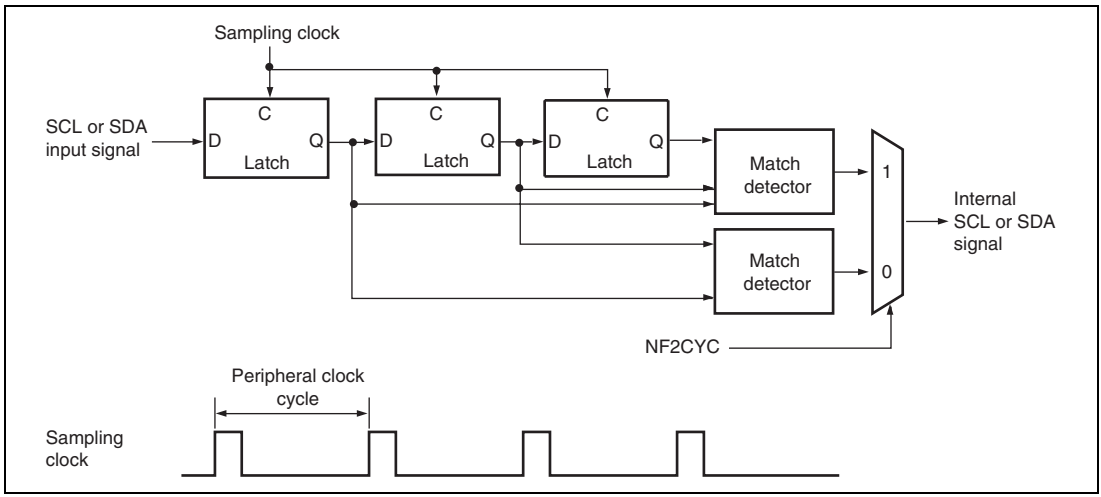


Figure 19.17 Block Diagram of Noise Filter

19.4.8 Using the IICRST Bit to Reset I²C Bus Interface 3

Some registers and the control part for I²C of the I²C bus interface 3 can be reset by writing 1 to the IICRST bit in ICCR2. Figure 19.18 shows an example of the sequence for resetting the I²C bus interface 3 by using the IICRST bit.

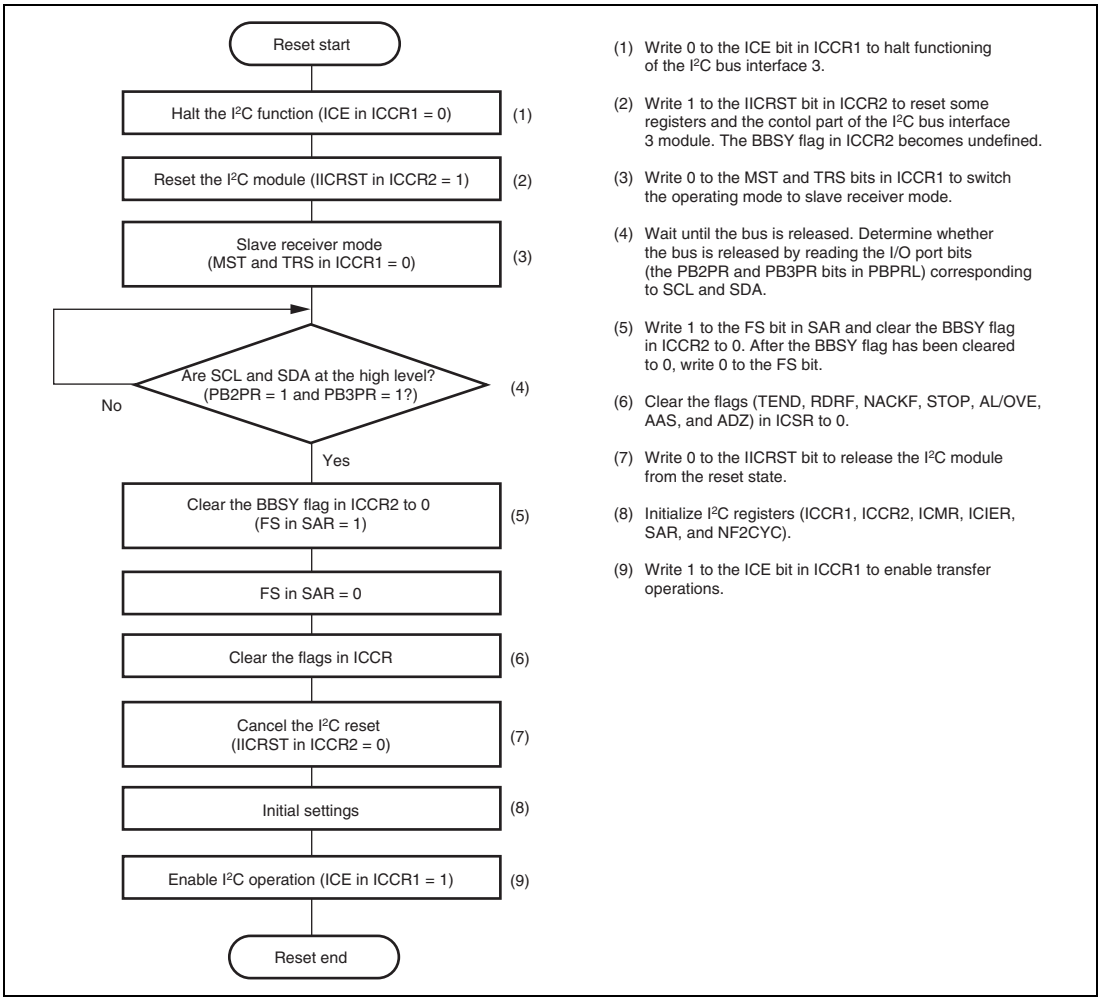


Figure 19.18 Sequence for Using the IICRST Bit to Reset I²C Bus Interface 3

19.4.9 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 19.19 to 19.22.

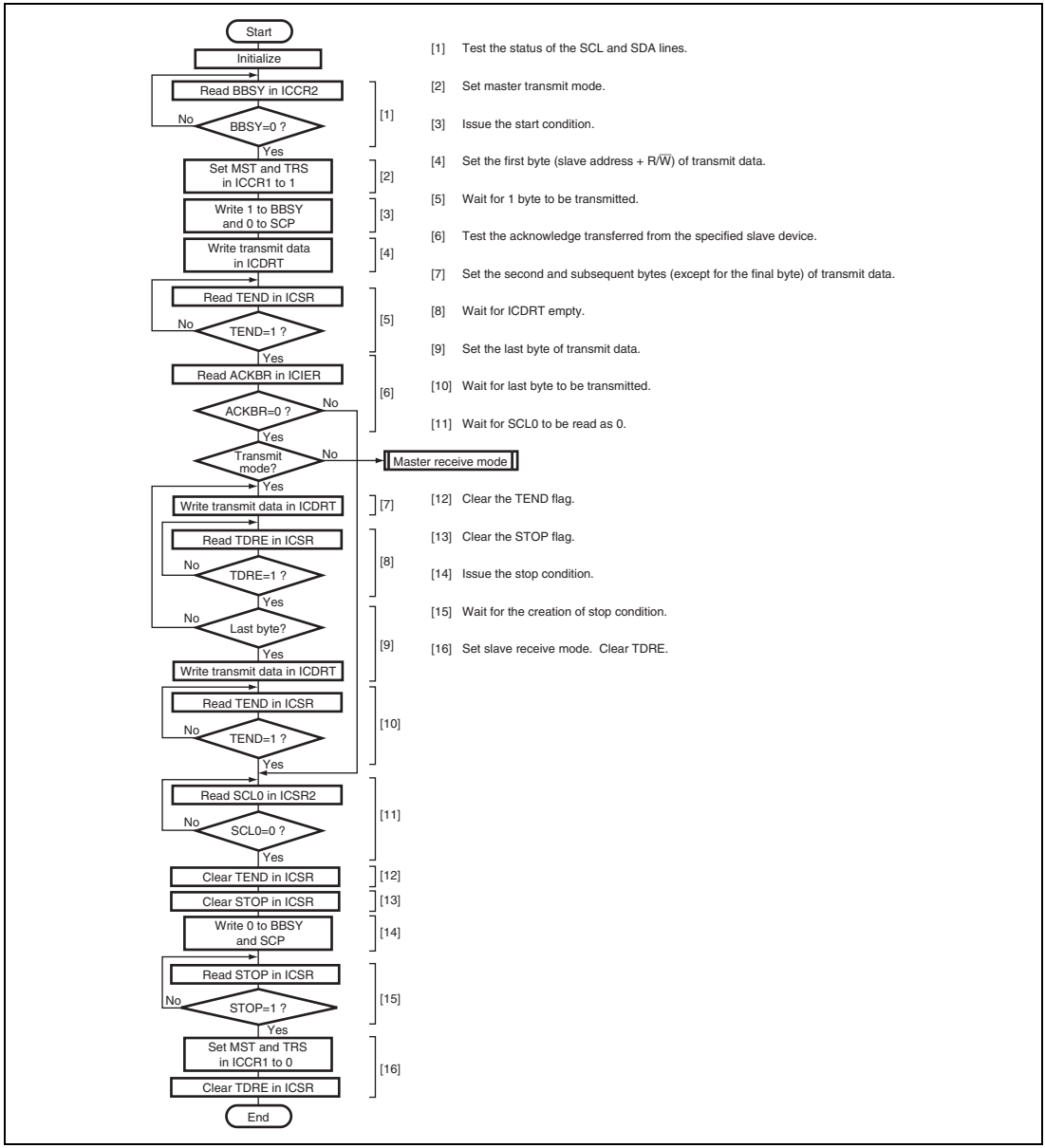
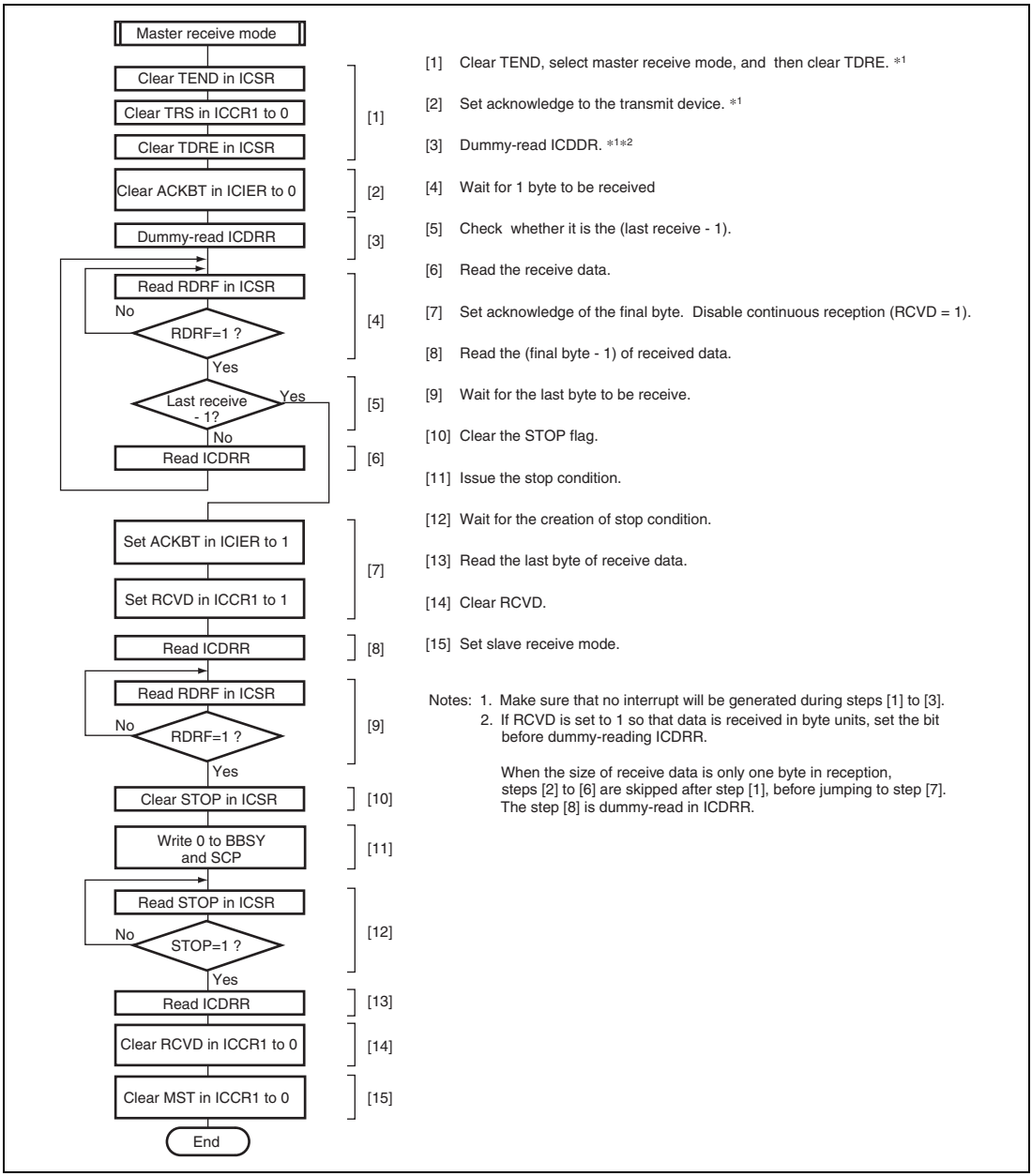


Figure 19.19 Sample Flowchart for Master Transmit Mode



[1] Clear TEND, select master receive mode, and then clear TDRE. *1

[2] Set acknowledge to the transmit device. *1

[3] Dummy-read ICDDR. *1*2

[4] Wait for 1 byte to be received

[5] Check whether it is the (last receive - 1).

[6] Read the receive data.

[7] Set acknowledge of the final byte. Disable continuous reception (RCVD = 1).

[8] Read the (final byte - 1) of received data.

[9] Wait for the last byte to be receive.

[10] Clear the STOP flag.

[11] Issue the stop condition.

[12] Wait for the creation of stop condition.

[13] Read the last byte of receive data.

[14] Clear RCVD.

[15] Set slave receive mode.

Notes: 1. Make sure that no interrupt will be generated during steps [1] to [3].
2. If RCVD is set to 1 so that data is received in byte units, set the bit before dummy-reading ICDRR.

When the size of receive data is only one byte in reception, steps [2] to [6] are skipped after step [1], before jumping to step [7]. The step [8] is dummy-read in ICDRR.

Figure 19.20 Sample Flowchart for Master Receive Mode

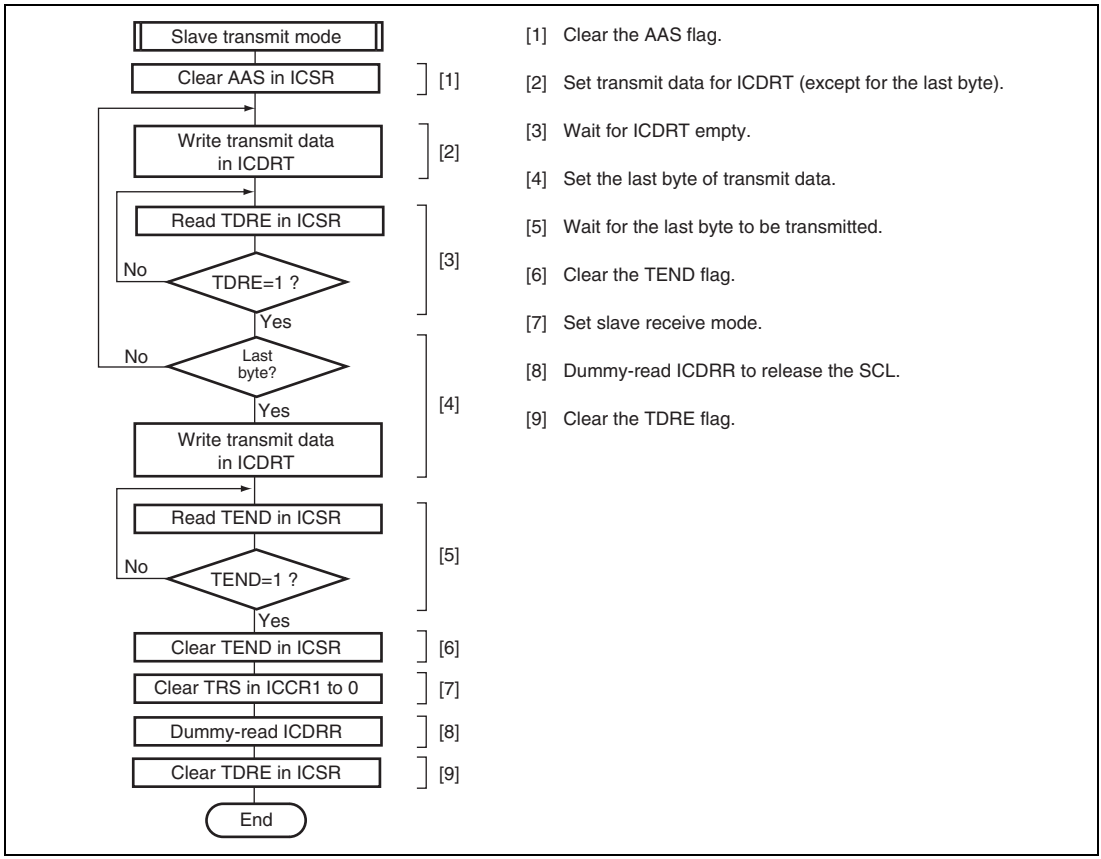


Figure 19.21 Sample Flowchart for Slave Transmit Mode

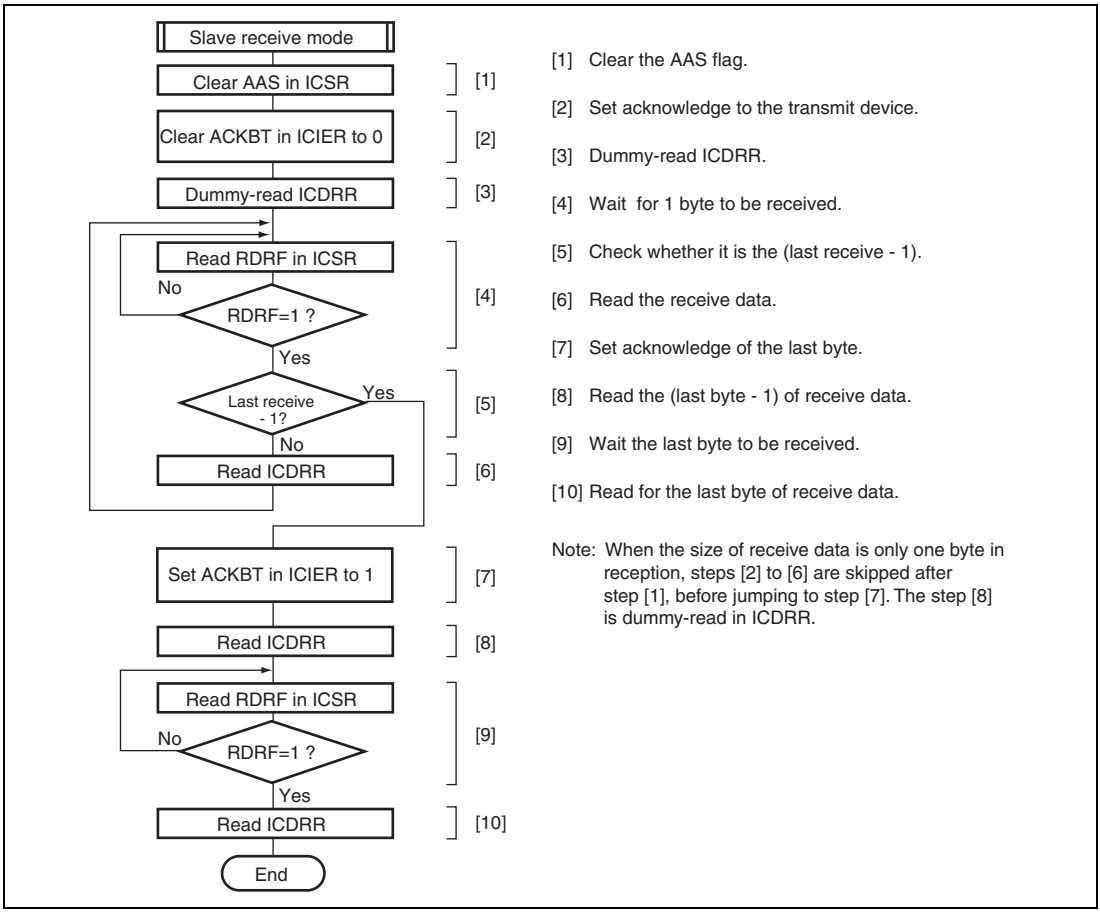


Figure 19.22 Sample Flowchart for Slave Receive Mode

19.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost/overrun error. Table 19.4 shows the contents of each interrupt request.

Table 19.4 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I^2C Bus Format	Clocked Synchronous Serial Format
Transmit data Empty	TXI	(TDRE = 1) • (TIE = 1)	√	√
Transmit end	TEI	(TEND = 1) • (TEIE = 1)	√	√
Receive data full	RXI	(RDRF = 1) • (RIE = 1)	√	√
STOP recognition	STPI	(STOP = 1) • (STIE = 1)	√	—
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	√	—
Arbitration lost/ overrun error			√	√

When the interrupt condition described in table 19.4 is 1, the CPU executes an interrupt exception handling. Note that a TXI or RXI interrupt can activate the DMAC or DTC if the setting for DMAC or DTC activation has been made. In such a case, an interrupt request is not sent to the CPU. Interrupt sources should be cleared in the exception handling. The TDRE and TEND bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data is written to ICDRT. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

19.6 Data Transfer Using DTC

In the I^2C bus format, the slave device and transfer direction are selected through the slave address and R/$\overline{W}$ bit, and data reception is confirmed and the last frame is indicated through the acknowledge bit. Therefore, when the DTC is used to transfer data continuously, the DTC processing should be done in combination with the CPU processing activated by interrupts.

Table 19.5 shows an example of I^2C data transfer using the DTC. This example assumes that the transfer data count is determined in advance in slave mode.

Table 19.5 Example of Data Transfer Using DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/W bit transmit/receive	Transmitted by DTC (ICDR writing)	Transmitted by CPU (ICDR writing)	Received by CPU (ICDR reading)	Received by CPU (ICDR reading)
Dummy data read	—	Processed by CPU (ICDR writing)	—	—
Main data transmit/receive	Transmitted by DTC (ICDR writing)	Received by DTC (ICDR reading)	Transmitted by DTC (ICDR writing)	Received by DTC (ICDR reading)
Last frame processing	Not necessary	Received by CPU (ICDR reading)	Not necessary	Received by CPU (ICDR reading)
DTC transfer data frame count setting	Transmission: Actual data count + 1 (+1 is required for the slave address + R/W bit transfer)	Reception; Actual data count	Transmission; Actual data count	Reception; Actual data count

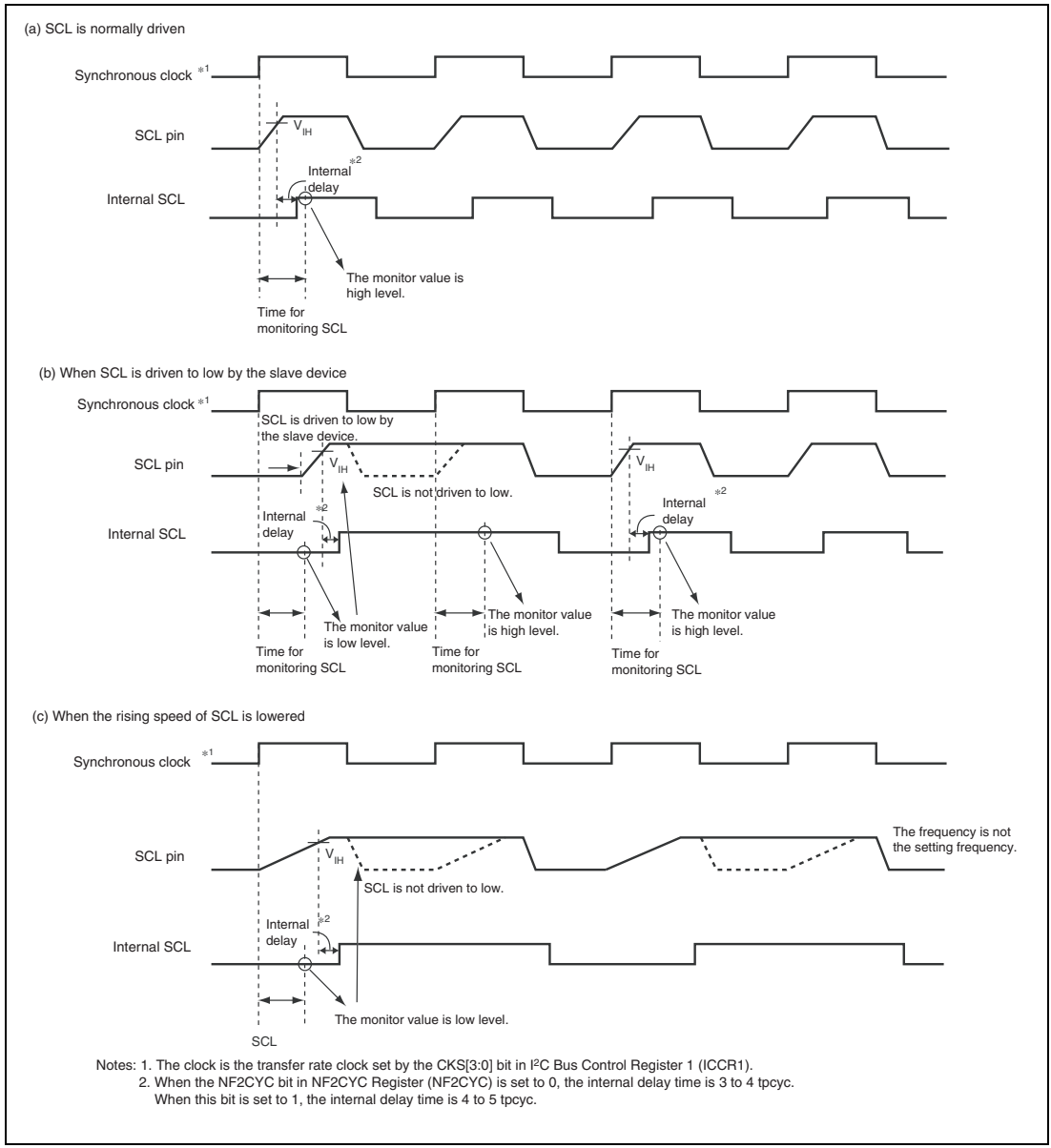
19.7 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 19.23 shows the timing of the bit synchronous circuit and table 19.6 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.



Notes: 1. The clock is the transfer rate clock set by the CKS[3:0] bit in I²C Bus Control Register 1 (ICCR1).
2. When the NF2CYC bit in NF2CYC Register (NF2CYC) is set to 0, the internal delay time is 3 to 4 tpcyc. When this bit is set to 1, the internal delay time is 4 to 5 tpcyc.

Figure 19.23 Bit Synchronous Circuit Timing

Table 19.6 Time for Monitoring SCL

CKS[3]	CKS[2]	Time for Monitoring SCL
0	0	9 tpcyc*
	1	21 tpcyc*
1	0	39 tpcyc*
	1	87 tpcyc*

Note: * tpcyc indicates peripheral clock (Pϕ) cycle.

19.8 Usage Notes

19.8.1 Setting for Multi-Master Operation

In multi-master operation, when the setting for IIC transfer rate (ICCR1.CKS[3:0]) makes this LSI slower than the other masters, pulse cycles with an unexpected length will infrequently be output on SCL.

Be sure to specify a transfer rate that is at least 1/1.8 of the fastest transfer rate among the other masters.

19.8.2 Note on Master Receive Mode

Reading ICDRR around the falling edge of the 8th clock might fail to fetch the receive data.

In addition, when RCVD is set to 1 around the falling edge of the 8th clock and the receive buffer is full, a stop condition may not be issued.

Use either of the following measures 1 or 2 against the situations above.

1. In master receive mode, read ICDRR before the rising edge of the 8th clock.
2. In master receive mode, set RCVD to 1 so that data is received in byte units.

19.8.3 Note on Setting ACKBT in Master Receive Mode

In master receive mode operation, set ACKBT before the falling edge of the 8th SCL cycle of the last data being continuously transferred. Not doing so can lead to an overrun for the slave transmission device.

19.8.4 Note on the States of Bits MST and TRN when Arbitration Is Lost

When sequential bit-manipulation instructions are used to set the MST and TRS bits to select master transmission in multi-master operation, a conflicting situation where AL in ICSR = 1 but the mode is master transmit mode (MST = 1 and TRS = 1) may arise; this depends on the timing of the loss of arbitration when the bit manipulation instruction for TRS is executed.

This can be avoided in either of the following ways.

- In multi-master operation, use the MOV instruction to set the MST and TRS bits.
- When arbitration is lost, check whether the MST and TRS bits are 0. If the MST and TRS bits have been set to a value other than 0, clear the bits to 0.

19.8.5 Access to ICE and IICRST Bits during I²C Bus Operations

Writing 0 to the ICE bit in ICCR1 or 1 to the IICRST bit in ICCR2 while this LSI is in any of the following states (1 to 4) causes the BBSY flag in ICCR2 and the STOP flag in ICSR to become undefined.

1. This module is the I²C bus master in master transmit mode (MST = 1 and TRS = 1 in ICCR1).
2. This module is the I²C bus master in master receive mode (MST = 1 and TRS = 0 in ICCR1).
3. This module is transmitting data in slave transmit mode (MST = 0 and TRS = 1 in ICCR1).
4. This module is transmitting acknowledge signals in slave receive mode (MST = 0 and TRS = 0 in ICCR1).

Executing any of the following procedures releases the BBSY flag in ICCR2 from the undefined state.

- Input a start condition (falling edge of SDA while SCL is at the high level) to set the BBSY flag to 1.
- Input a stop condition (rising edge of SDA while SCL is at the high level) to clear the BBSY flag to 0.
- If the module is in master transmit mode, issue a start condition by writing 1 and 0 to the BBSY flag and the SCP bit in ICCR2, respectively, while SCL and SDA are at the high level. The BBSY flag is set to 1 on output of the start condition (falling edge of SDA while SCL is at the high level).

- With the module in master transmit or master receive mode, SDA at the low level, and no other device holding SCL at the low level, issue a stop condition by writing 0 to the BBSY flag and the SCP bit in ICCR2. The BBSY flag is cleared to 0 on output of the stop condition (rising edge of SDA while SCL is at the high level).
- Writing 1 to the FS bit in SAR clears the BBST flag to 0.

19.8.6 Using the IICRST Bit to Initialize the Registers

- Writing 1 to the IICRST bit sets the SDAO and SCLO bits in ICCR2 to 1.
- Writing 1 to the IICRST bit in master transmit mode or slave transmit mode sets the TDRE flag in ICSR to 1.
- During a reset due to the IICRST bit being set to 1, writing to the BBSY flag and the SCP and SDAO bits is invalid.
- Even during a reset due to the IICRST bit being set to 1, the input of a start (falling edge of SDA while SCL is at the high level) or stop (rising edge of SDA while SCL is at the high level) condition on SCL and SDA causes the BBSY flag to be set to 1 or cleared to 0, respectively.

19.8.7 Operation of I^2C Bus Interface 3 while ICE = 0

Writing 0 to the ICE bit in ICCR1 disables output on SCL and SDA. However, input on SCL and SDA remains valid. This module operates in accord with the signals input on SCL and SDA.

19.8.8 Note on Master Transmit Mode

When the ACKE bit is set to 1 in master transmit mode, issue a stop condition after confirming the falling edge of the 9th clock of SCL.

Section 20 A/D Converter (ADC)

This LSI includes a successive approximation type 12-bit A/D converter.

20.1 Features

- 12-bit resolution
- Input channels: Eight channels
- High-speed conversion

  When Aϕ = 50 MHz: Minimum 1.0 µs per channel

  AD clock = 50 MHz, 50 conversion states
- Two operating modes
  - Single-cycle scan mode: Continuous A/D conversion on one to four channels
  - Continuous scan mode: Repetitive A/D conversion on one to four channels
- Eight A/D data registers

  A/D conversion results are stored in 16-bit A/D data registers (ADDR) that correspond to the input channels.
- Sample-and-hold function

  Sample-and-hold circuits are built into the A/D converter of this LSI, simplifying the configuration of the external analog input circuitry. Multiple channels can be sampled simultaneously because sample-and-hold circuits can be dedicated to channels 0 to 2.
  - Group A (GrA): Analog input pins selected from channels 0, 1, and 2 can be simultaneously sampled.
- Three methods for starting A/D conversion

  Software: Setting of the ADST bit in ADCR

  Timer: TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2

  TRGAN, TRG4AN, and TRG4BN from the MTU2S

  External trigger: $\overline{\text{ADTRG}}$ (LSI pin)
- Selectable analog input channel

  A/D conversion of a selected channel is accomplished by setting the A/D analog input channel select registers (ADANSR).
- A/D conversion end interrupt, DMAC transfer function, and DTC transfer function are supported

  On completion of A/D conversion, A/D conversion end interrupts (ADI) can be generated and the DMAC or DTC can be activated by an ADI.

Figure 20.1 shows a block diagram of the A/D converter.

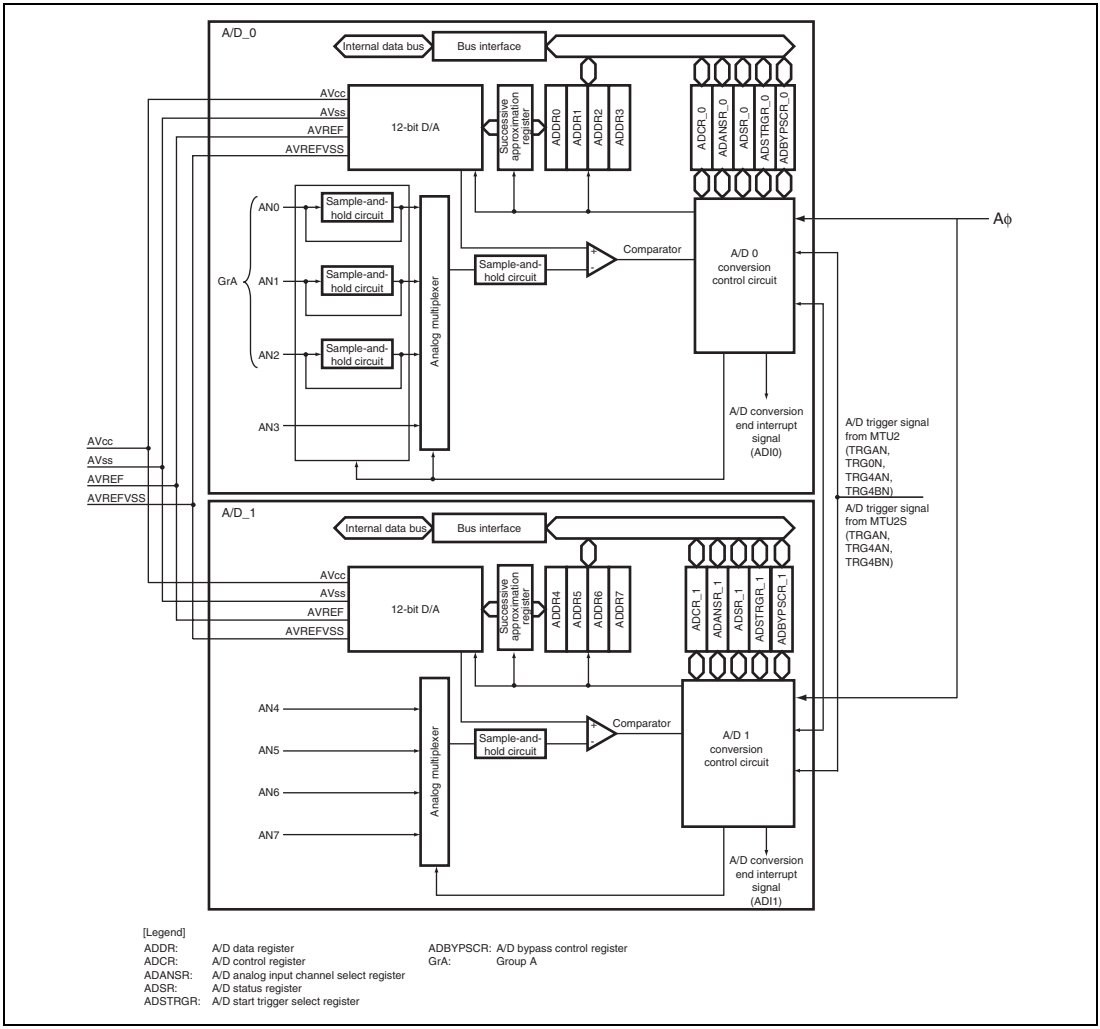


Figure 20.1 Block Diagram of A/D Converter

20.2 Input/Output Pins

Table 20.1 shows the configuration of the pins used by the A/D converter. For the pin usage, refer to the usage notes in section 20.7, Usage Notes.

Table 20.1 Pin Configuration

Module	Pin Name	I/O	Function
Common	AV_{CC}	Input	Analog block power supply pin
	AV_{SS}	Input	Analog block ground pin
	AVREF	Input	Analog block reference power supply pin (high)
	AVREFVSS	Input	Analog block reference power supply pin (low)
	$\overline{\text{ADTRG}}$	Input	A/D external trigger input pin
A/D module 0 (A/D_0)	AN0	Input	Analog input pin 0 (Group A)
	AN1	Input	Analog input pin 1 (Group A)
	AN2	Input	Analog input pin 2 (Group A)
	AN3	Input	Analog input pin 3
A/D module 1 (A/D_1)	AN4	Input	Analog input pin 4
	AN5	Input	Analog input pin 5
	AN6	Input	Analog input pin 6
	AN7	Input	Analog input pin 7

20.3 Register Descriptions

The A/D converter has the following registers.

Table 20.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D control register_0	ADCR_0	R/W	H'00	H'FFFFE800	8
A/D status register_0	ADSR_0	R/W	H'00	H'FFFFE802	8
A/D start trigger select register_0	ADSTRGR_0	R/W	H'00	H'FFFFE81C	8
A/D analog input channel select register_0	ADANSR_0	R/W	H'00	H'FFFFE820	8
A/D bypass control register_0	ADBYPSCR_0	R/W	H'00	H'FFFFE830	8
A/D data register 0	ADDR0	R	H'0000	H'FFFFE840	16
A/D data register 1	ADDR1	R	H'0000	H'FFFFE842	16
A/D data register 2	ADDR2	R	H'0000	H'FFFFE844	16
A/D data register 3	ADDR3	R	H'0000	H'FFFFE846	16
A/D control register_1	ADCR_1	R/W	H'00	H'FFFFEC00	8
A/D status register_1	ADSR_1	R/W	H'00	H'FFFFEC02	8
A/D start trigger select register_1	ADSTRGR_1	R/W	H'00	H'FFFFEC1C	8
A/D analog input channel select register_1	ADANSR_1	R/W	H'00	H'FFFFEC20	8
A/D bypass control register_1	ADBYPSCR_1	R/W	H'00	H'FFFFEC30	8
A/D data register 4	ADDR4	R	H'0000	H'FFFFEC40	16
A/D data register 5	ADDR5	R	H'0000	H'FFFFEC42	16
A/D data register 6	ADDR6	R	H'0000	H'FFFFEC44	16
A/D data register 7	ADDR7	R	H'0000	H'FFFFEC46	16

20.3.1 A/D Control Registers 0 and 1 (ADCR_0 and ADCR_1)

ADCR is an 8-bit readable/writable register that selects A/D conversion mode and others.

Bit:	7	6	5	4	3	2	1	0
	ADST	ADCS	ACE	ADIE	-	-	TRGE	EXTRG
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*1	R/W*2	R/W*2	R/W*2	R	R	R/W*2	R/W*2

Notes: 1. Do not overwrite 1 while the ADST bit is set to 1.
2. Do not modify the value of this bit while the ADST bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	ADST	0	R/W	A/D Start When this bit is cleared to 0, A/D conversion is stopped and the A/D converter enters the idle state. When this bit is set to 1, A/D conversion is started. In single-cycle scan mode, this bit is automatically cleared to 0 when A/D conversion ends on the selected single channel. In continuous scan mode, A/D conversion is continuously performed for the selected channels in sequence until this bit is cleared by software, a reset, or in software standby mode. Note: Setting of the ADST bit must be done while it is cleared to 0 to prevent incorrect operations.
6	ADCS	0	R/W	A/D Continuous Scan Selects either a single-cycle or a continuous scan in scan mode. This bit is valid only when scan mode is selected. 0: Single-cycle scan 1: Continuous scan When changing the operating mode, first clear the ADST bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	ACE	0	R/W	Automatic Clear Enable Enables or disables the automatic clearing of ADDR after ADDR is read by the CPU or DMAC. When this bit is set to 1, ADDR is automatically cleared to H'0000 after the CPU or DMAC reads ADDR. This function allows the detection of any renewal failures of ADDR. 0: Automatic clearing of ADDR after being read is disabled. 1: Automatic clearing of ADDR after being read is enabled.
4	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the generation of A/D conversion end interrupts (ADI) to the CPU. Operating modes must be changed when the ADST bit is 0 to prevent incorrect operations. When A/D conversion ends and the ADF bit in ADSR is set to 1 and this bit is set to 1, ADI is sent to the CPU. By clearing the ADF bit or the ADIE bit to 0, ADI can be cleared. In addition, ADIE activates the DMAC when an ADI is generated. At this time, no interrupt to the CPU is generated. 0: Generation of A/D conversion end interrupt is disabled 1: Generation of A/D conversion end interrupt is enabled
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1	TRGE	0	R/W	Trigger Enable Enables or disables A/D conversion start by the external trigger input ($\overline{\text{ADTRG}}$) or A/D conversion start triggers from the MTU2 and MTU2S (TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2 and TRGAN, TRG4AN, and TRG4BN from the MTU2S). For selection of the external trigger and A/D conversion start trigger from the MTU2 or MTU2S, see the description of the EXTRG bit. 0: A/D conversion start by the external trigger or an A/D conversion start trigger from the MTU or MTU2S is disabled 1: A/D conversion start by the external trigger or an A/D conversion start trigger from the MTU2 or MTU2S is enabled
0	EXTRG	0	R/W	Trigger Select Selects the external trigger ($\overline{\text{ADTRG}}$) or an A/D conversion start trigger from the MTU2 or MTU2S as an A/D conversion start trigger. When the external trigger is selected (EXTRG = 1), upon input of a low-level pulse to the $\overline{\text{ADTRG}}$ pin after the TRGE bit is set to 1, the A/D converter detects the falling edge of the pulse, and sets the ADST bit in ADCR to 1. The operation which is performed when 1 is written to the ADST bit by software is subsequently performed. A/D conversion start by the external trigger input is enabled only when the ADST bit is cleared to 0. When the external trigger is used as an A/D conversion start trigger, the low-level pulse input to the $\overline{\text{ADTRG}}$ pin must be at least 1.5 Pϕ clock cycles in width. 0: A/D converter is started by the A/D conversion start trigger from the MTU2 or MTU2S 1: A/D converter is started by the external pin ($\overline{\text{ADTRG}}$)

20.3.2 A/D Status Registers 0 to 1 (ADSR_0 and ADSR_1)

ADSR is an 8-bit readable/writable register that indicates the status of the A/D converter.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADF
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way. Do not overwrite 0 while this flag is 0.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADF	0	R/(W)*	A/D End Flag A status flag that indicates the completion of A/D conversion. [Setting condition] • When A/D conversion on all specified channels is completed in scan mode [Clearing conditions] • When 0 is written after reading ADF = 1 • When the DMAC is activated by an ADI interrupt and ADDR is read

20.3.3 A/D Start Trigger Select Registers 0 and 1 (ADSTRGR_0 and ADSTRGR_1)

ADSTRGR selects an A/D conversion start trigger from the MTU2 or MTU2S. The A/D conversion start trigger is used as an A/D conversion start source when the TRGE bit in ADCR is set to 1 and the EXTRG bit in ADCR is set to 0.

Bit:	7	6	5	4	3	2	1	0
	-	STR6	STR5	STR4	STR3	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	STR6	0	R/W	Start Trigger 6 Enables or disables the A/D conversion start request input from the MTU2S. 0: Disables the A/D conversion start by TRGAN trigger (MTU2S). 1: Enables the A/D conversion start by TRGAN trigger (MTU2S).
5	STR5	0	R/W	Start Trigger 5 Enables or disables the A/D conversion start request input from the MTU2S. 0: Disables the A/D conversion start by TRG4AN trigger (MTU2S). 1: Enables the A/D conversion start by TRG4AN trigger (MTU2S).
4	STR4	0	R/W	Start Trigger 4 Enables or disables the A/D conversion start request input from the MTU2S. 0: Disables the A/D conversion start by TRG4BN trigger (MTU2S). 1: Enables the A/D conversion start by TRG4BN trigger (MTU2S).

Bit	Bit Name	Initial Value	R/W	Description
3	STR3	0	R/W	Start Trigger 3 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRG0N trigger (MTU2). 1: Enables the A/D conversion start by TRG0N trigger (MTU2).
2	STR2	0	R/W	Start Trigger 2 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRGAN trigger (MTU2). 1: Enables the A/D conversion start by TRGAN trigger (MTU2).
1	STR1	0	R/W	Start Trigger 1 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRG4AN trigger (MTU2). 1: Enables the A/D conversion start by TRG4AN trigger (MTU2).
0	STR0	0	R/W	Start Trigger 0 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRG4BN trigger (MTU2). 1: Enables the A/D conversion start by TRG4BN trigger (MTU2).

20.3.4 A/D Analog Input Channel Select Registers 0 and 1 (ADANSR_0 and ADANSR_1)

ADANSR is an 8-bit readable/writable register that selects an analog input channel.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ANS3	ANS2	ANS1	ANS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	ANS3	0	R/W	Setting bits in the A/D analog input channel select register to 1 selects a channel that corresponds to a specified bit. For the correspondence between analog input pins and bits, see table 20.3. When changing the analog input channel, the ADST bit in ADCR must be cleared to 0 to prevent incorrect operations.
2	ANS2	0	R/W	
1	ANS1	0	R/W	
0	ANS0	0	R/W	

Table 20.3 Channel Select List

	Analog Input Channels	
Bit Name	**A/D_0**	**A/D_1**
ANS0	AN0	AN4
ANS1	AN1	AN5
ANS2	AN2	AN6
ANS3	AN3	AN7

20.3.5 A/D Bypass Control Registers 0 and 1 (ADBYPSCR_0 and ADBYPSCR_1)

For A/D conversion of group A (GrA), it can be selected whether or not to use the sample-and-hold circuits dedicated to the group A channels.

Setting the SH bit in ADBYPSCR_0 to 1 selects the sample-and-hold circuits dedicated to the channels. When the sample-and-hold circuits are not to be used, the A/D conversion time does not include the time for sampling in the dedicated sample-and-hold circuits. For details, refer to section 20.4, Operation.

The function of the SH bit in this register is available only for A/D converter_0. A/D converter_1 is always in the same state as when the SH bit is set to 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SH
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SH	0	R/W	Dedicated Sample-and-Hold Circuit Select (ADBYPSCR_0 only) 0: Does not select the sample-and-hold circuits 1: Selects the sample-and-hold circuits This bit is a reserved bit in ADBYPSCR_1. The writing value should always be 0.

20.3.6 A/D Data Registers 0 to 7 (ADDR0 to ADDR7)

ADDRs are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. (See table 20.4.)

The converted 12-bit data is stored in bits 11 to 0.

The initial value of ADDR is H'0000.

After ADDR is read, ADDR can be automatically cleared to H'0000 by setting the ACE bit in ADCR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	ADD[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	—	All 0	R	Reserved
11 to 0	ADD[11:0]	All 0	R	12-bit data

Table 20.4 Correspondence between Analog Channels and Registers (ADDR0 to ADDR11)

Analog Input Channels	A/D Data Registers
AN0	ADDR0
AN1	ADDR1
AN2	ADDR2
AN3	ADDR3
AN4	ADDR4
AN5	ADDR5
AN6	ADDR6
AN7	ADDR7

20.4 Operation

The A/D converter has two operating modes: single-cycle scan mode and continuous scan mode. In single-cycle scan mode, A/D conversion is performed once on one or more specified channels and then it ends. In continuous scan mode, the A/D conversion is performed sequentially on one or more specified channels until the ADST bit is cleared to 0.

The ADCS bit in the A/D control register (ADCR) is used to select the operating mode. Setting the ADCS bit to 0 selects single-cycle scan mode and setting the ADCS bit to 1 selects continuous scan mode. In both modes, A/D conversion starts on the channel with the lowest number in the analog input channels selected by the A/D analog input channel select register (ADANSR) from AN0 to AN3.

In single-cycle scan mode, when one cycle of A/D conversion on all specified channels is completed, the ADF bit in ADSR is set to 1 and the ADST bit is automatically cleared to 0. In continuous scan mode, when conversion on all specified channels is completed, the ADF bit in ADSR is set to 1. To stop A/D conversion, write 0 to the ADST bit. When the ADF bit is set to 1, if the ADIE bit in ADCR is set to 1, an A/D conversion end interrupt (ADI) is generated. When clearing the ADF bit to 0, read the ADF bit while set to 1 and then write 0. However, when the DMAC or DTC is activated by an ADI interrupt, the ADF bit is automatically cleared to 0.

20.4.1 Single-Cycle Scan Mode

The following example shows the operation when analog input channels 0 to 3 (AN0 to AN3) are selected and the A/D conversion is performed in single-cycle scan mode using four channels.

1. Set the ADCS bit in the A/D control register (ADCR) to 0.
2. Set all bits ANS0 to ANS3 in the A/D analog input channel select register (ADANSR) to 1.
3. Set the SH bit in the A/D bypass control register_0 (ADBYPSCR_0).
4. Set the ADST bit in the A/D control register (ADCR) to 1 to start A/D conversion.
5. Channels 0 to 2 (GrA) are sampled simultaneously*. Then, A/D conversion is performed on channel 0. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR0. In the same way, channels 1 and 2 are converted and the A/D conversion results are transferred to ADDR1 and ADDR2.
6. A/D conversion of channel 3 is then started. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR3.

7. When A/D conversion ends on all specified channels (AN0 to AN3), the ADF bit is set to 1, the ADST bit is automatically cleared to 0, and the A/D conversion ends. At this time, if the ADIE bit is set to 1, an ADI interrupt is generated after the A/D conversion.

Note: * The operation depends on the SH bit setting in ADBYPSCR_0. For details, see figures 20.2 and 20.3.

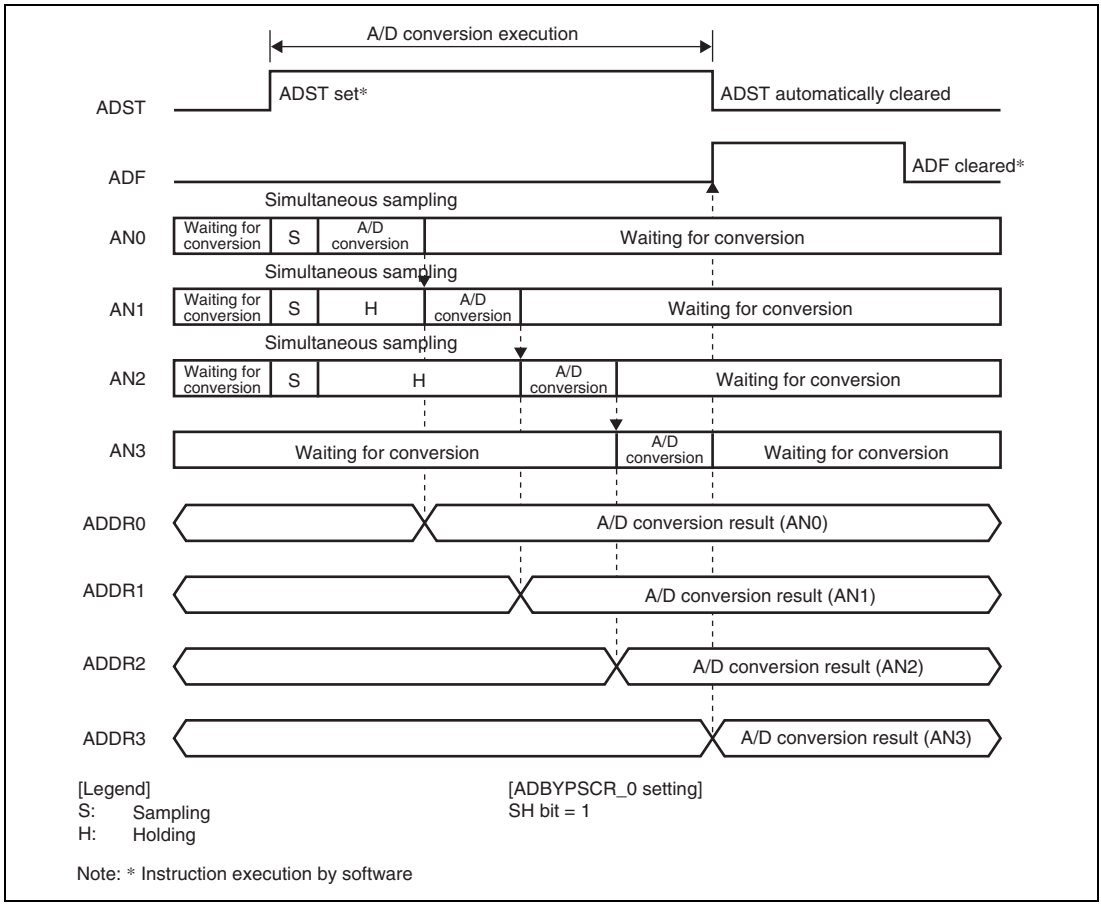


Figure 20.2 Example 1 of A/D_0 Converter Operation (Single-Cycle Scan Mode and Sample-and-Hold Circuit Enabled)

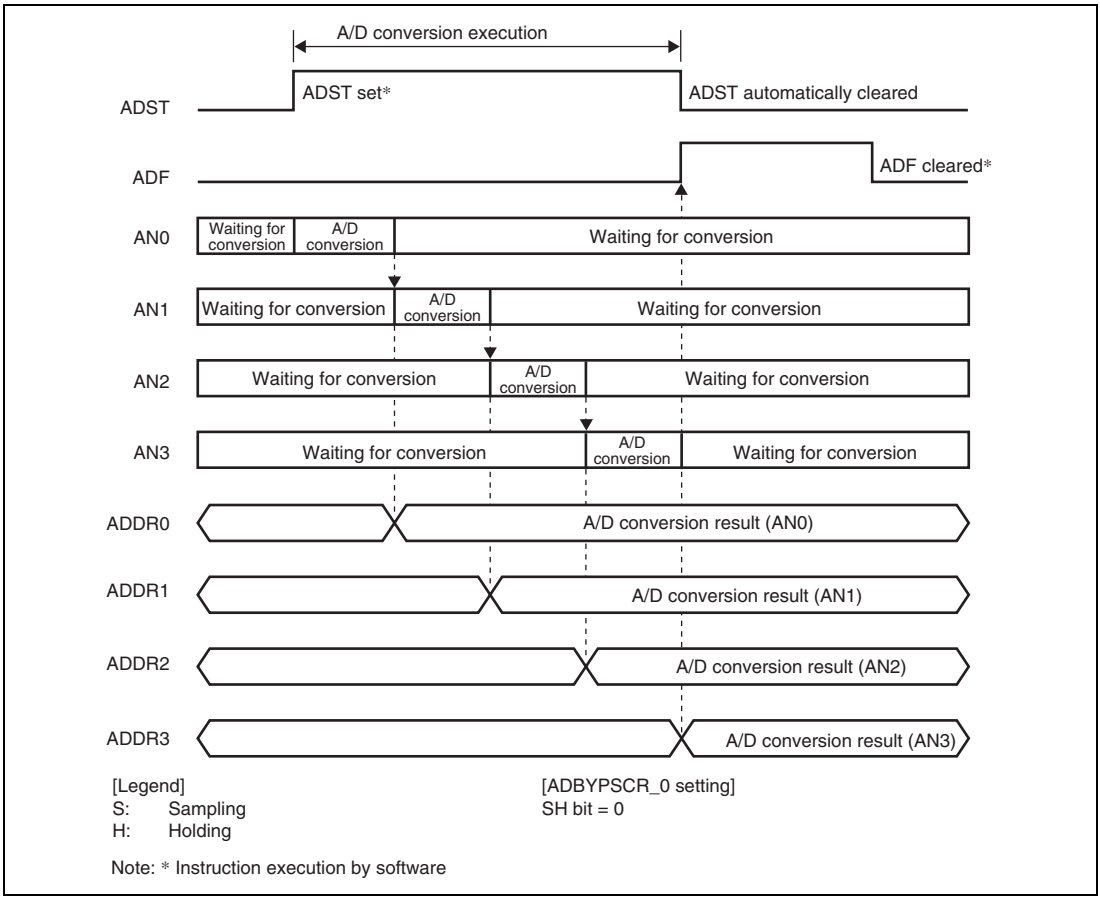


Figure 20.3 Example 2 of A/D_0 Converter Operation (Single-Cycle Scan Mode and Sample-and-Hold Circuit Disabled)

20.4.2 Continuous Scan Mode

The following example shows the operation when analog input 0, 2, and 3 (AN0, AN2, AN3) are selected and the A/D conversion is performed in continuous scan mode using the three channels. This operation also applies to the A/D_1 conversion.

1. Set the ADCS bit in the A/D control register (ADCR) to 0.
2. Set all bits of ANS0, ANS2, and ANS3 in the A/D analog input channel select register (ADANSR) to 1.
3. Set the SH bit in the A/D bypass control register_0 (ADBYPSCR_0).
4. Set the ADST bit in the A/D control register (ADCR) to 1 to start A/D conversion.
5. Channels 0 and 2 (GrA) are sampled simultaneously*. As the ANS1 bit in ADANSR is set to 0, channel 1 is not sampled. Then the A/D conversion on channel 0 is started. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR0. In the same way, channel 2 is converted and the A/D conversion result is transferred to ADDR2. The A/D conversion is not performed on channel 1.
6. The A/D conversion of channel 3 starts. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR3.
7. When the A/D conversion ends on all the specified channels (AN0, AN2, and AN3), the ADF bit is set to 1. At this time, if the ADIE bit is set to 1, an ADI interrupt is generated after the A/D conversion.
8. Steps 5 to 7 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, the A/D conversion stops. After this, if the ADST bit is set to 1, the A/D conversion starts again and repeats steps 5 to 7.

Note: * The operation depends on the SH bit setting in ADBYPSCR_0. For details, see figures 20.4 and 20.5.

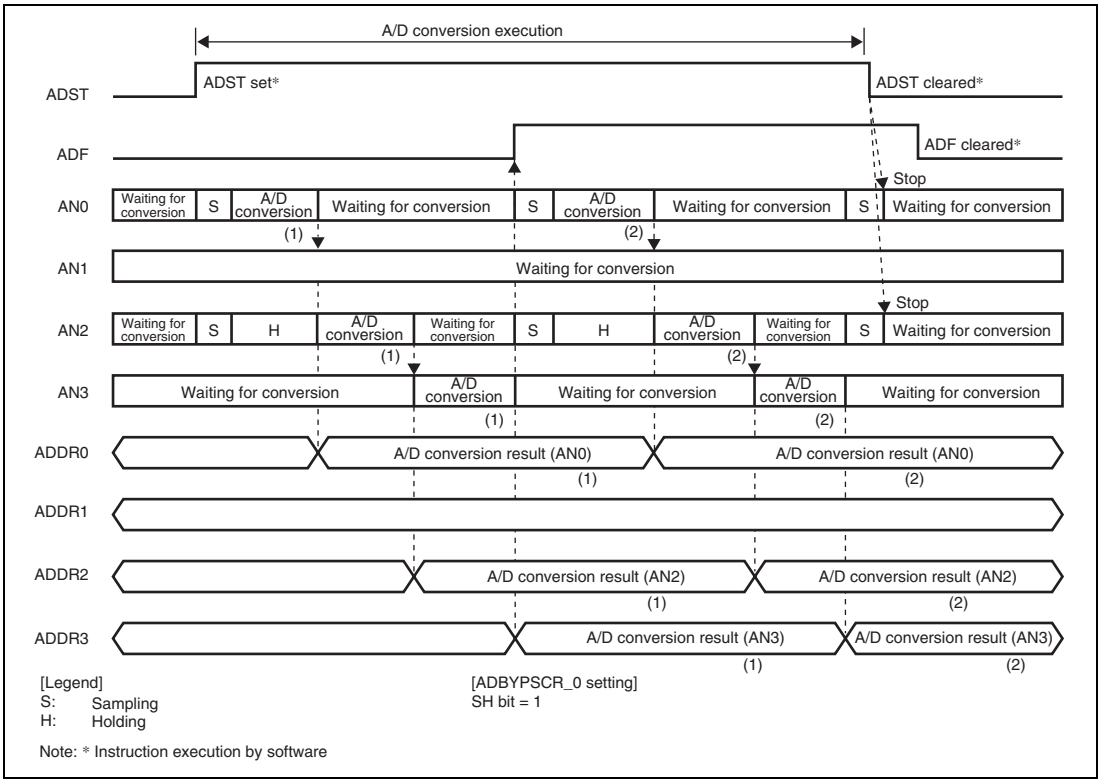


Figure 20.4 Example 1 of A/D_0 Converter Operation (Continuous Scan Mode and Sample-and-Hold Circuit Enabled)

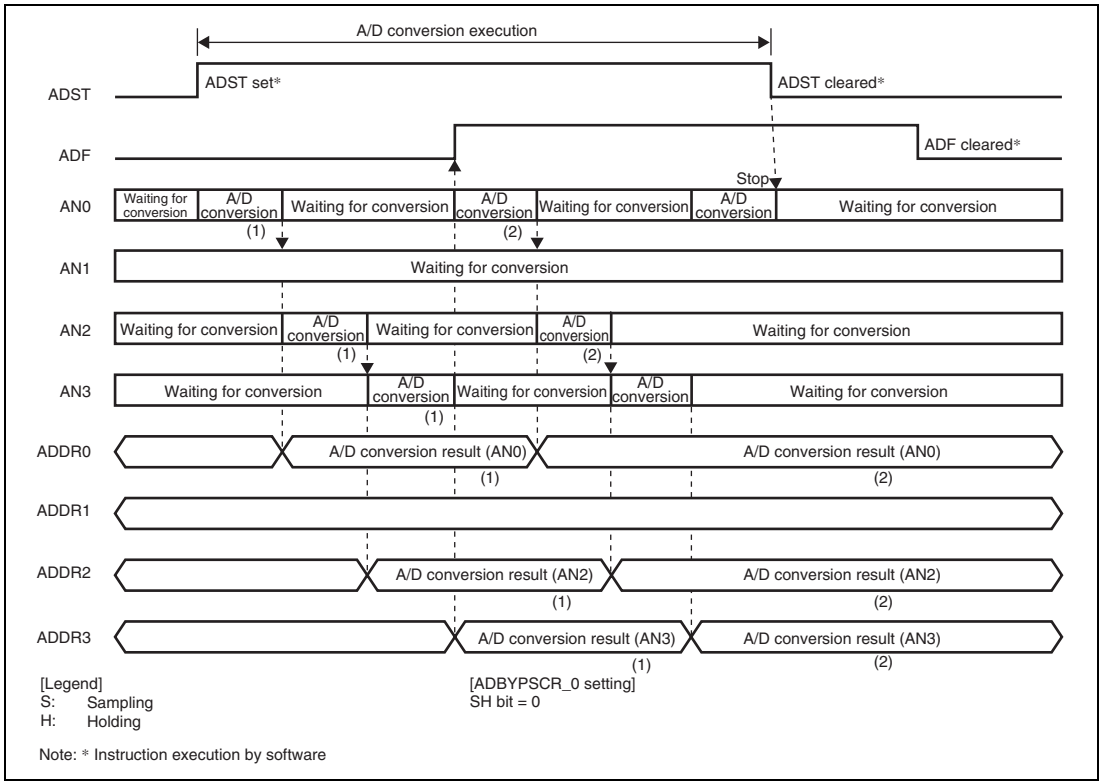


Figure 20.5 Example 2 of A/D_0 Converter Operation (Continuous Scan Mode and Sample-and-Hold Circuit Disabled)

20.4.3 Input Sampling and A/D Conversion Time

The A/D converter has built-in sample-and-hold circuits. Channels 0 to 2 can be simultaneously sampled as one group when the SH bit in ADBYPSCR_0 is set to 1. This group is referred to as Group A (GrA) (in table 20.5). When the SH bit is cleared to 0, these channels are sampled individually in the same way as other channels.

Setting the ADST bit to 1 starts A/D conversion. The A/D conversion time (t_{CONV}) from the beginning to the end of conversion is determined by the following four time factors (figure 20.6): the A/D conversion start delay time (t_D), sampling time (t_{SPLSH}), sampling time (t_{SPL}), and A/D conversion processing time; the A/D conversion time (t_{CONV}) is the sum of these times. t_{SPLSH} can be reduced according to the following procedure.

To reduce t_{SPLSH}, clear the SH bit in ADBYPSCR_0 to 0 (initial value). Note that when GrA channels should be sampled simultaneously, the SH bit should be set to 1 to provide appropriate t_{SPLSH}. t_{SPLSH} indicates the time required for the operation of the sample-and-hold circuits dedicated to channels 0 to 2 and it does not depend on the number of channels sampled simultaneously.

In continuous scan mode, the A/D conversion time (t_{CONV}) given in table 20.6 applies to the conversion time of the first cycle. The conversion time of the second and subsequent cycles is expressed as ($t_{CONV} - t_D + 6$).

Table 20.6 shows the state for the Aϕ1 clock. The value is calculated by multiplying the cycle time of Aϕ and the number of the state. The Aϕ should always be set to Pϕ or greater (Pϕ ≤ Aϕ) value.

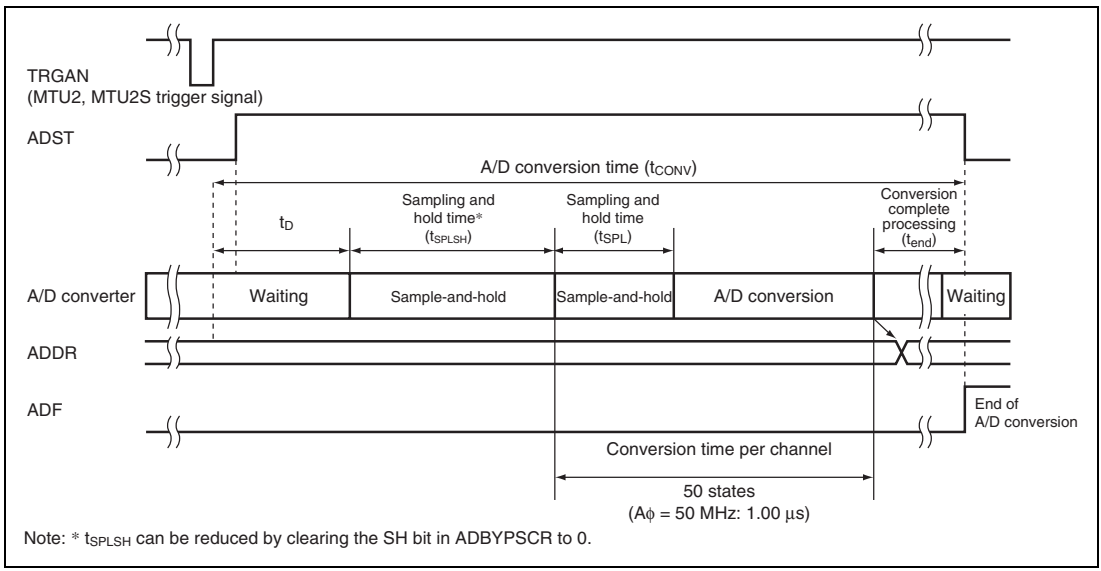
Table 20.5 Correspondence between Analog Input Channels and Groups being Allowed Simultaneous Sampling

A/D Converter Module	Analog Input Channels	Group
A/D converter module 0	AN0	GrA
	AN1	
	AN2	
	AN3	—
A/D converter module 1	AN4	—
	AN5	—
	AN6	—
	AN7	—

Table 20.6 A/D Conversion Time

Item		Symbol	Min.	Typ.	Max.
A/D conversion start delay time		t_D	11*[1]	—	15*[2]
Analog input sampling time of sample-and-hold circuits dedicated to GrA		t_{SPLSH}	—	30	—
Analog input sampling time of sample-and-hold circuit common to all channels		t_{SPL}	—	20	—
Completion of conversion		t_{end}	—	4	—
A/D conversion time	ADBYPSCR.SH = 0	t_{CONV}	50n + 15*[3]	—	50n + 19*[3]
	ADBYPSCR.SH = 1		50n + 45*[3]	—	50n + 49*[3]

Notes: 1. A/D activation by MTU2, MTU2S trigger signal
2. A/D activation by the external trigger signal
3. n is a number of channel (n = 1 to 4)



Note: * t_{SPLSH} can be reduced by clearing the SH bit in ADBYPSCR to 0.

Figure 20.6 A/D Conversion Timing

20.4.4 A/D Converter Activation by MTU2 and MTU2S

A/D conversion is activated by the A/D conversion start triggers (TRGAN, TRG0N, TRG4N, and TRG4BN) from the MTU2 and A/D conversion start triggers (TRGAN, TRG4AN, and TRG4BN) from the MTU2S. To enable this function, set the TRGE bit in ADCR to 1 and clear the EXTRG bit to 0. After this setting is made, if an A/D conversion start trigger from the MTU2 or MTU2S is generated, the ADST bit is set to 1. The time between the setting of the ADST bit to 1 and the start of the A/D conversion is the same as when A/D conversion is activated by writing 1 to the ADST bit by software.

20.4.5 External Trigger Input Timing

The A/D conversion can also be externally triggered. To input an external trigger, set the pin function controller (PFC) to select the $\overline{\text{ADTRG}}$ pin function, drive the $\overline{\text{ADTRG}}$ pin high, set the TRGE bit to 1 in ADCR, clear the ADST bit to 0, and set the EXTRG bit to 1. In this state, input a trigger through the $\overline{\text{ADTRG}}$ pin. A falling edge of the $\overline{\text{ADTRG}}$ signal sets the ADST bit to 1 in ADCR, starting the A/D conversion. Other operations are conducted in the same way as when A/D conversion is activated by writing 1 to the ADST bit by software. Figure 20.7 shows the timing.

The ADST bit is set to 1 after ((5 – n*)Pϕ) states have elapsed from the point at which the A/D converter detects a falling edge on the $\overline{\text{ADTRG}}$ pin.

Notes: * n = 0 when Pϕ : Aϕ = 1:1
n = 1 when Pϕ : Aϕ = 1:2
n = 2 when Pϕ : Aϕ = 1:4

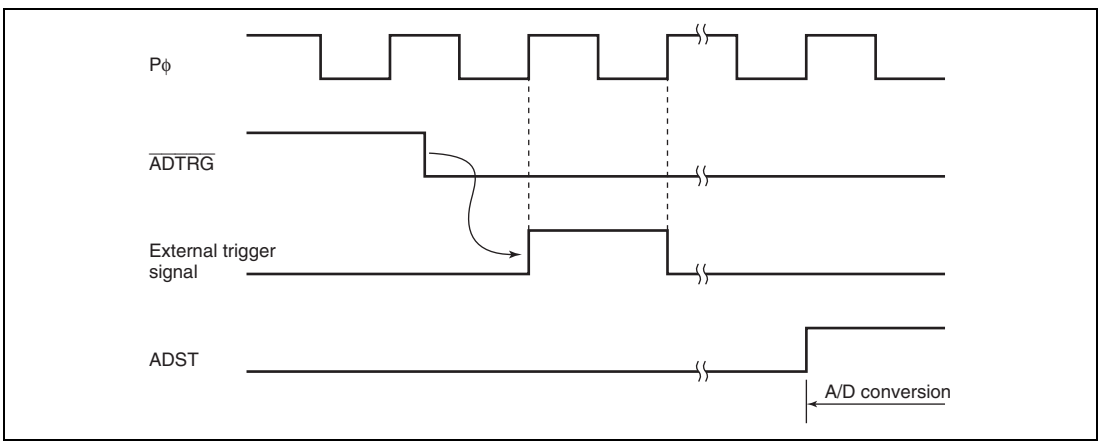


Figure 20.7 External Trigger Input Timing

20.4.6 Example of ADDR Auto-Clear Function

When the A/D data register (ADDR) is read by the CPU or DMAC, ADDR can be automatically cleared to H'0000 by setting the ACE bit in ADCR to 1. This function allows the detection of non-updated ADDR states.

Figure 20.8 shows an example of when the auto-clear function of ADDR is disabled (normal state) and enabled.

When the ACE bit is 0 (initial value) and the A/D conversion result (H'0222) is not written to ADDR for some reason, the old data (H'0111) becomes the ADDR value. In addition, when the ADDR value is read into a general register using an A/D conversion end interrupt, the old data (H'0111) is stored in the general register. To detect a renewal failure, every time the old data needs to be stored in the RAM, a general register, etc.

When the ACE bit is 1, reading ADDR = H'0111 by the CPU, DMAC, or DTC automatically clears ADDR to H'0000. After this, if the A/D conversion result (H'0222) cannot be transferred to ADDR for some reason, the cleared data (H'0000) remains as the ADDR value. When this ADDR value is read into a general register, H'0000 is stored in the general register. Just by checking whether the read data value is H'0000 or not allows the detection of non-updated ADDR states.

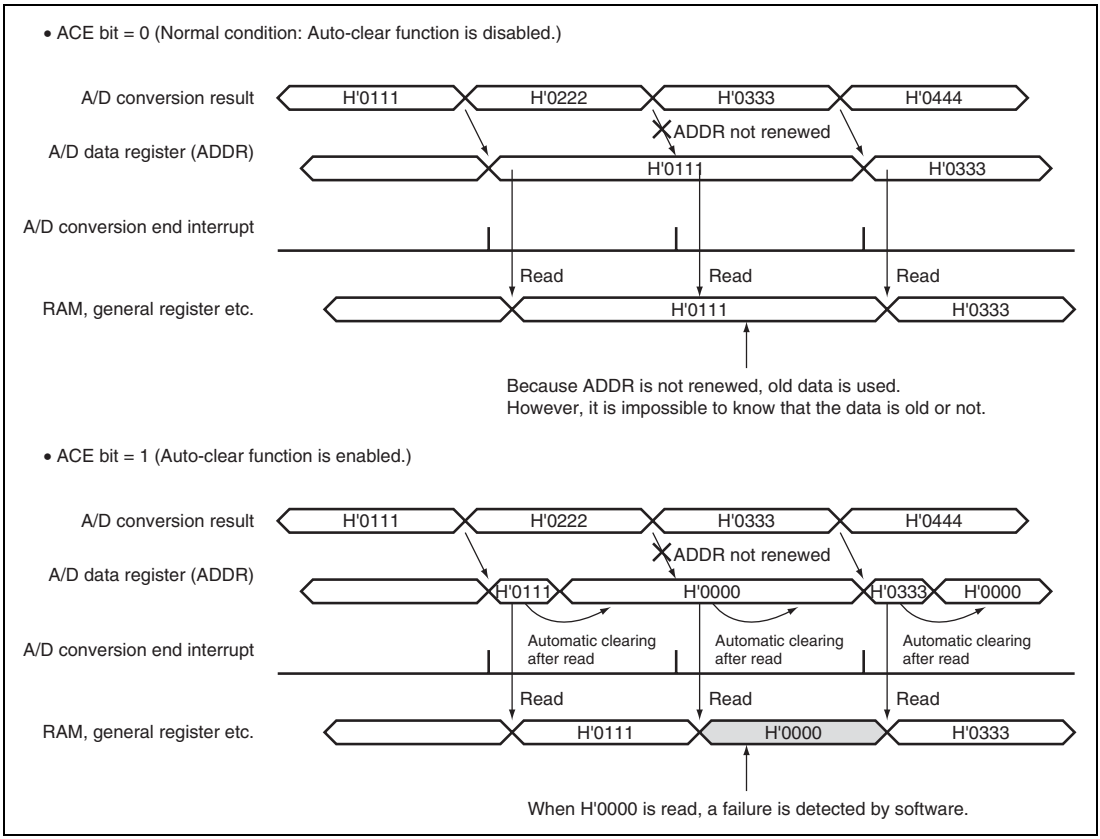


Figure 20.8 Example of When ADDR Auto-clear Function is Disabled (Normal Condition)/Enabled

20.5 Interrupt Sources and DMAC or DTC Transfer Requests

The A/D converter generates A/D conversion end interrupts (ADI). An ADI interrupt generation is enabled when the ADIE bit in ADCR is set to 1. The DMAC or DTC can be activated by the DMAC or DTC setting when an ADI interrupt is generated. At this time, no interrupt to the CPU is generated. When the DMAC or DTC is activated by an ADI interrupt, the ADF bit in ADSR is automatically cleared at the data transfer by the DMAC or DTC.

Table 20.7 AD Interrupt Sources

A/D Converter Module	Name	DMAC Activation Request	DTC Activation Request
A/D converter module 0	ADI0	Available	Available
A/D converter module 1	ADI1	Not available	Available

20.6 Definitions of A/D Conversion Accuracy

This LSI's A/D conversion accuracy definitions are given below.

- Resolution

  The number of A/D converter digital conversion output codes
- Offset error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic when the digital output value changes from the minimum voltage value (zero voltage) B'000000000000 to B'000000000001. Does not include a quantization error (see figure 20.9).
- Full-scale error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic when the digital output value changes from B'111111111110 to the maximum voltage value (full-scale voltage) B'111111111111. Does not include a quantization error (see figure 20.9).
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 20.9).
- Nonlinearity error

  The deviation of the actual A/D conversion characteristic from the ideal A/D conversion characteristic between zero voltage and full-scale voltage. Does not include offset error, full-scale error, or quantization error (see figure 20.9).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

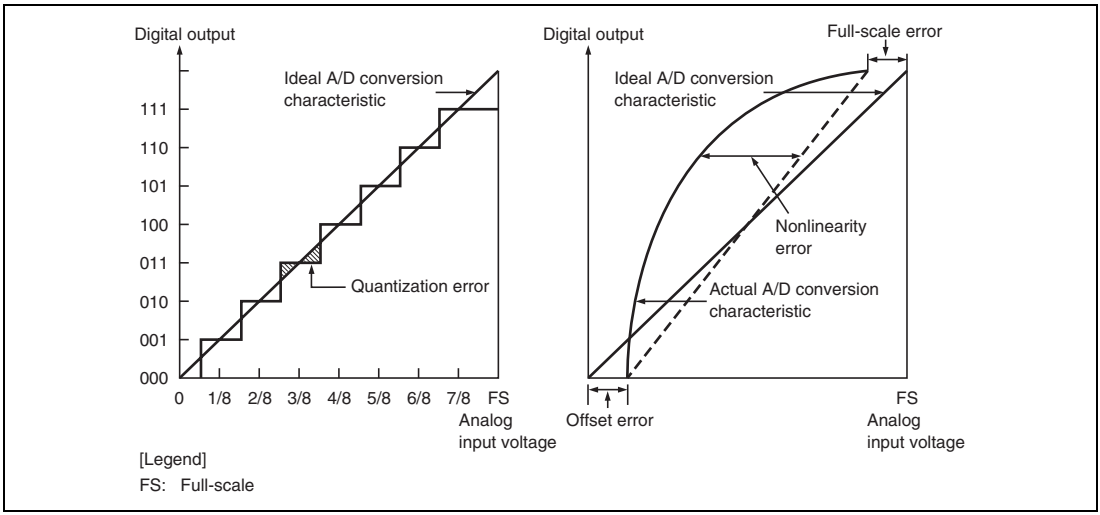


Figure 20.9 Definitions of A/D Conversion Accuracy

20.7 Usage Notes

20.7.1 Analog Input Voltage Range

The voltage applied to analog input pin (ANn) during A/D conversion should be in the range AVss ≤ ANn (n = 0 to 7) ≤ AVref.

20.7.2 Relationship between AVcc, AVss and VccQ, Vss

When using the A/D converter, set AVcc = 5.0 V ± 0.5 V and AVss = Vss. When the A/D converter is not used, set VccQ ≤ AVcc ≤ 5.0V ± 0.5 V, AVss = Vss, and do not leave the AVcc pin open.

20.7.3 Range of AVREF Pin Settings

Set AVREF = 4.5 V to AVcc when using the A/D converter, or set AVREF = AVcc when not using the A/D converter. Set AVREFVSS = AVSS, and do not leave the AVREFVSS pin open. If these conditions are not met, the reliability of the LSI may be adversely affected.

20.7.4 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided as much as possible. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values.

In addition, digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference power supply (AVREF), the analog power supply (AVcc), and the analog ground (AVss). AVss should be connected at one point to a stable digital ground (Vss) on the board.

20.7.5 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN7) and analog reference power supply (AVREF), a protection circuit should be connected between the AVcc and AVss, as shown in figure 20.10. The bypass capacitors connected to AVREF and the filter capacitor connected to ANn should be connected to the AVREFVSS. The 0.1-µF capacitor in figure 20.10 should be placed close to the pin. If a filter capacitor is connected as shown in figure 20.10, the input currents at the analog input pin (ANn) are averaged, and an error may occur. Careful consideration is therefore required when deciding the circuit constants.

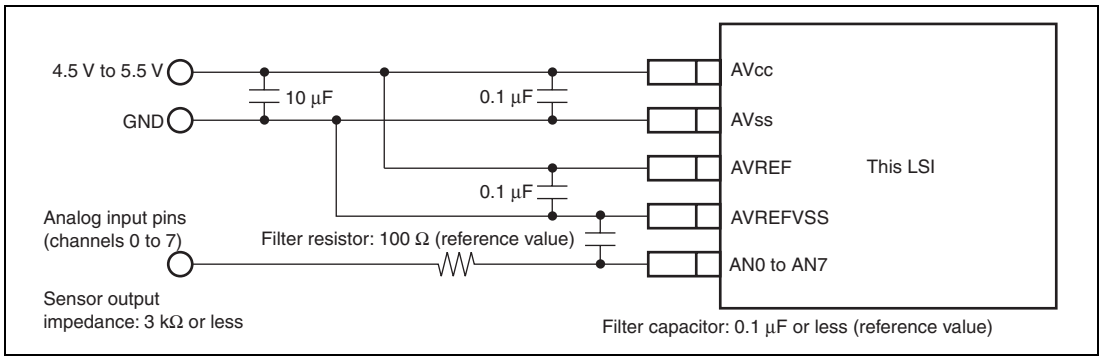


Figure 20.10 Example of Analog Input Pin Protection Circuit

20.7.6 Notes on Register Setting

- Set the ADST bit in the A/D control register (ADCR) after the A/D start trigger select register (ADSTRGR) and the A/D analog input channel select register (ADANSR) have been set.
- Do not modify the settings of the ADCS, ACE, ADIE, TRGE, and EXTRG bits while the ADST bit in the ADCR register is set to 1.
- Do not write 1 to the ADST bit while the ADST bit in the ADCR register is set to 1.
- Do not start the A/D conversion when the ANS bits (ANS[7:0]) in the A/D analog input channel select register (ADANSR) are all 0.

20.7.7 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 3 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 3 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. However, for A/D conversion in single mode with a large capacitance provided externally for A/D conversion in single mode, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater). When converting a high-speed analog signal or in scan mode, a low-impedance buffer should be inserted.

20.7.8 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

20.7.9 Notes when Two A/D Modules Run Simultaneously

This LSI has two A/D modules. When two modules run simultaneously, or if the conversion of the next A/D module is started during the conversion of the first A/D module, as shown in figures 20.11 and 20.12, the guaranteed absolute precision of the A/D conversion module which has been activated first will be the values as listed in tables 20.8 and 20.9. The absolute precision depends on the cycle difference ($T_{AD0\text{-}AD1}$ in figures 20.11 and 20.12) between the start of the first activated A/D conversion and the one of the next activated A/D conversion. Therefore, evaluate the specifications fully when two or more A/D modules are run simultaneously.

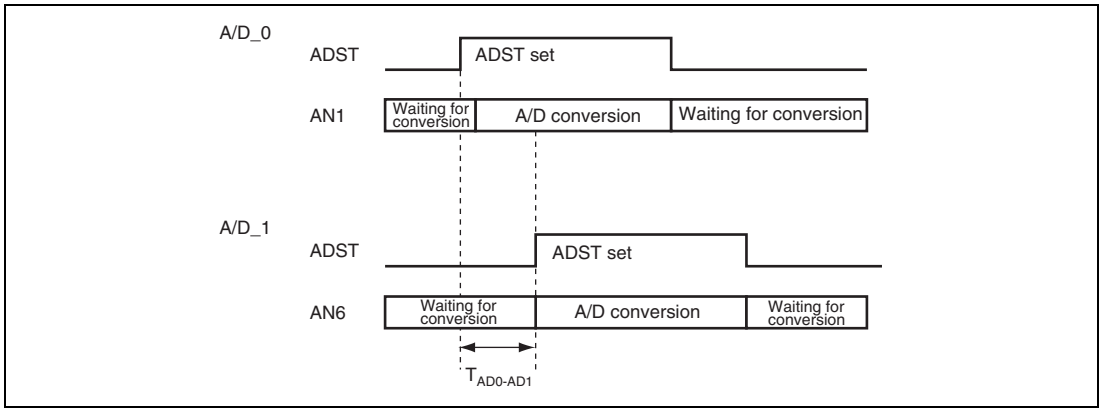


Figure 20.11 A/D Conversion Start Timing between A/D_0 Converter and A/D_1 Converter (Sample-and-Hold Circuits Disabled in A/D_0 and A/D_1)

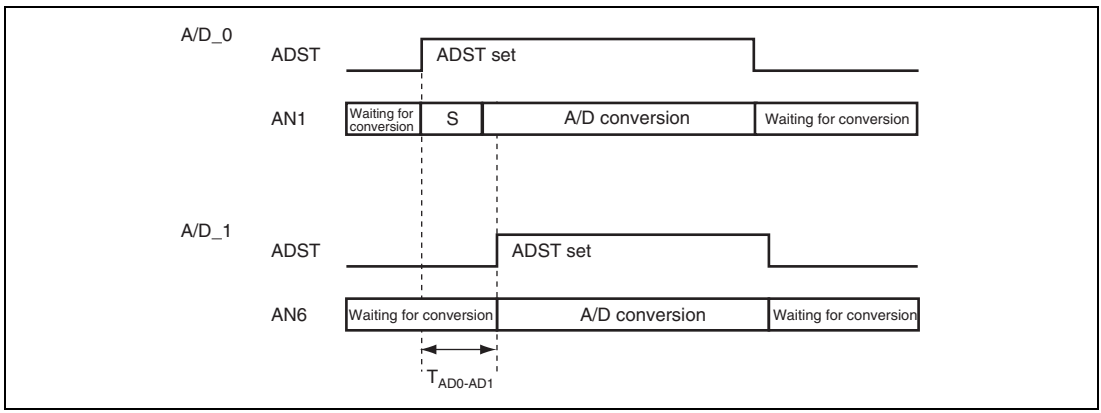


Figure 20.12 A/D Conversion Start Timing between A/D_0 Converter and A/D_1 Converter (Sample-and-Hold Circuit Enabled in A/D_0)

Table 20.8 Absolute Precision and A/D Conversion Start Cycle Difference, $T_{AD0\text{-}AD1}$ (Aϕ) between A/D_0 and A/D_1 in Figure 20.11

	$T_{AD0\text{-}AD1}$	Unit
	0 to 15, 21 to 30, 45 or more	**Aϕ (clock)**
Absolute precision	±8	LSB

Notes: 1. This table lists the A/D_0 absolute precision when the converter of A/D_0 is started first.
2. The precision of A/D_1 is ±8LSB regardless of $T_{AD0\text{-}AD1}$ when the converter of A/D_0 is started first.
3. When the conversion of A/D_0 and A/D_1 is started simultaneously, the absolute precision values of A/D_0 and A/D_1 are ±8LSB because $T_{AD0\text{-}AD1} = 0$.
4. When two A/D modules run simultaneously, the absolute precision of the first activated A/D is not guaranteed except for $T_{AD0\text{-}AD1}$.
5. When A/D_0 and A/D_1 are activated separately, each of $T_{AD0\text{-}AD1}$ values is 45 or more. Thus, the absolute precision values of A/D_0 and A/D_1 are ±8LSB.

Table 20.9 Absolute Precision and A/D Conversion Start Cycle Difference, $T_{AD0\text{-}AD1}$ (Aϕ) between A/D_0 and A/D_1 in Figure 20.12

	$T_{AD0\text{-}AD1}$	Unit
	0 to 15, 33 to 45, 55 to 65, 83 to 95, 107 or more	**Aϕ (clock)**
Absolute precision	±8	LSB

Notes: 1. This table lists the A/D_0 absolute precision when the converter of A/D_0 is started first.
2. The precision of A/D_1 is ±8LSB regardless of $T_{AD0\text{-}AD1}$ when the converter of A/D_0 is started first.
3. When the conversion of A/D_0 and A/D_1 is started simultaneously, the absolute precision values of A/D_0 and A/D_1 are ±8LSB because $T_{AD0\text{-}AD1} = 0$.
4. When two A/D modules run simultaneously, the absolute precision of the first activated A/D is not guaranteed except for $T_{AD0\text{-}AD1}$.
5. When A/D_0 and A/D_1 are activated separately, each of $T_{AD0\text{-}AD1}$ values is 107 or more. Thus, the absolute precision values of A/D_0 and A/D_1 are ±8LSB.

Section 21 Controller Area Network (RCAN-ET)

21.1 Summary

21.1.1 Overview

This document primarily describes the programming interface for the RCAN-ET module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-ET implementation can ensure the design is successful.

21.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-ET module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

21.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the RCAN-ET user interface LSI engineers must use this document to understand the hardware requirements.

21.1.4 References

1. CAN Licence Specification, Robert Bosch GmbH, 1992
2. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
3. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
4. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
5. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2003)

21.1.5 Features

- supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 16 Mailbox version
- Clock 20 to 50 MHz
- 15 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- programmable CAN data rate up to 1MBit/s
- transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- data buffer access without SW handshake requirement in reception
- flexible micro-controller interface
- flexible interrupt structure

21.2 Architecture

The RCAN-ET device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control and CAN Interface. The figure below shows the block diagram of the RCAN-ET Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

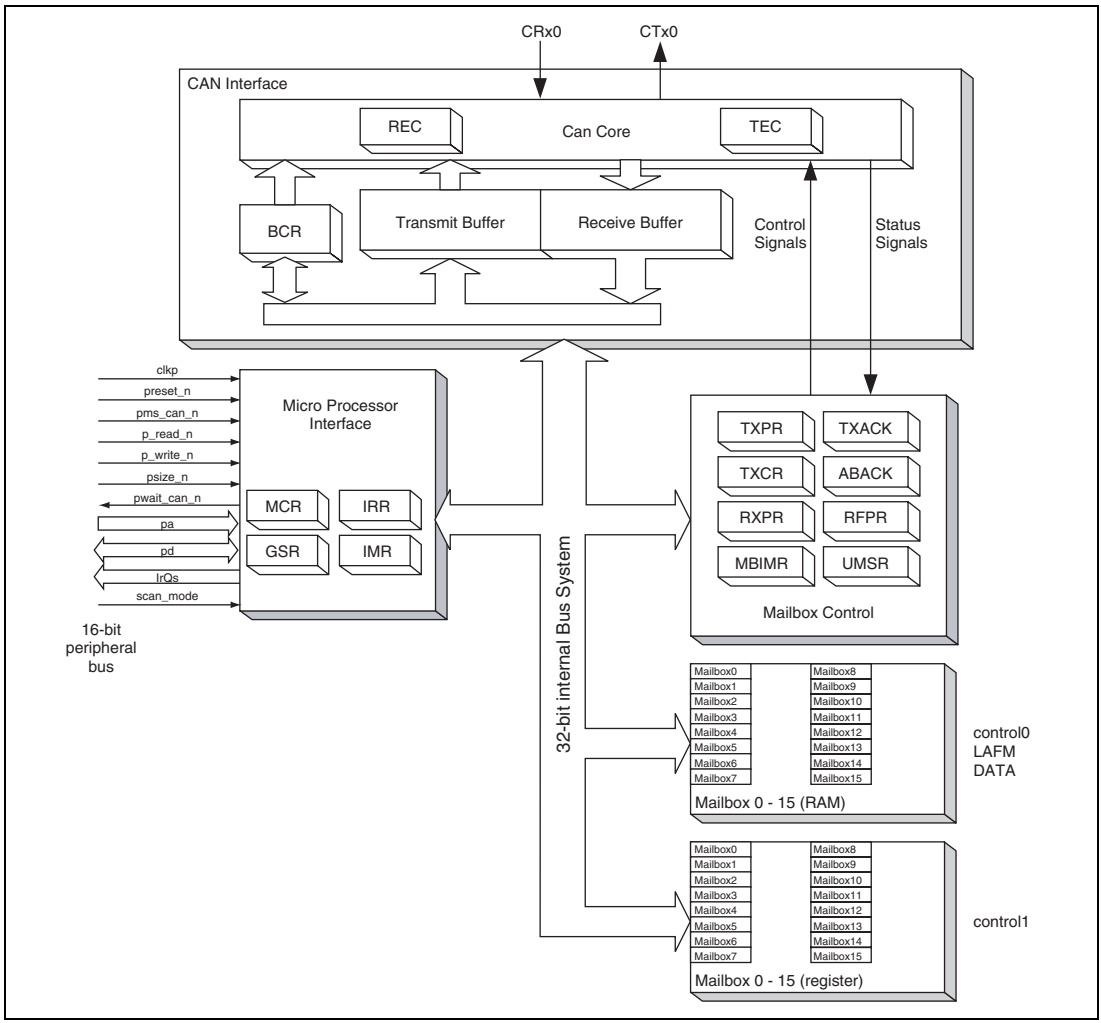


Figure 21.1 RCAN-ET Architecture

Important: Although core of RCAN-ET is designed based on a 32-bit bus system, the whole RCAN-ET including MPI for the CPU has 16-bit bus interface to CPU. In that case, LongWord (32-bit) access must be implemented as 2 consecutive word (16-bit) accesses. In this manual, LongWord access means the two consecutive accesses.

- Micro Processor Interface (MPI)

  The MPI allows communication between the Renesas CPU and RCAN-ET's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-ET so that the RCAN-ET can automatically exit the Sleep mode.

  It contains registers such as MCR, IRR, GSR and IMR.

- Mailbox

  The Mailboxes consists of RAM configured as message buffers and registers. There are 16 Mailboxes, and each mailbox has the following information.

  <RAM>

  — CAN message control (identifier, rtr, ide,etc)

  — CAN message data (for CAN Data frames)

  — Local Acceptance Filter Mask for reception

  <Registers>

  — CAN message control (dlc)

  — 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

- Mailbox Control

  The Mailbox Control handles the following functions:

  — For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

  — To transmit messages, RCAN-ET will run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly.

  — Arbitrates Mailbox accesses between the CPU and the Mailbox Control.

  — Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

- CAN Interface

  This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2, 4]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the

CAN Data Link Controller.

21.3 Programming Model - Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer. Please bear in mind that the user manual reports all settings allowed by the RCAN-ET IP. Different use of RCAN-ET is not allowed.

21.3.1 Memory Map

The diagram of the memory map is shown below.

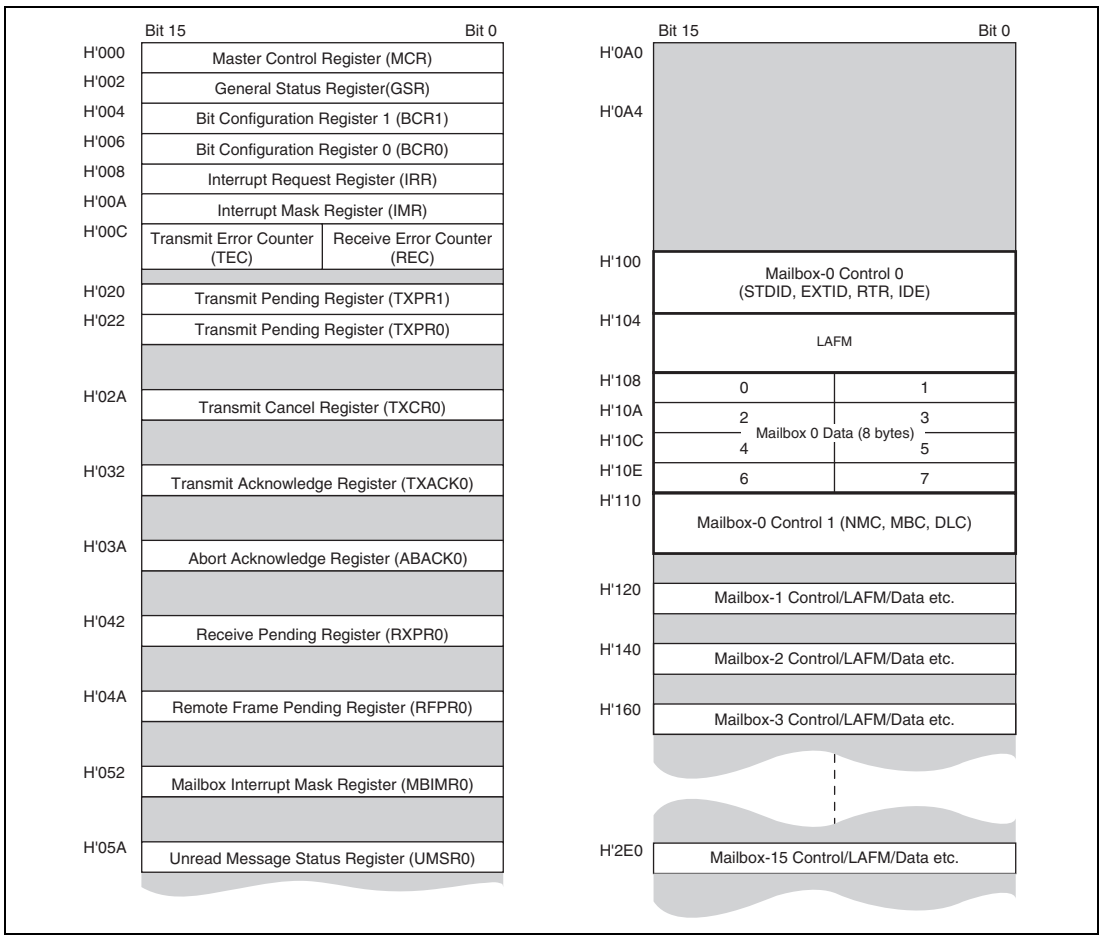


Figure 21.2 RCAN-ET Memory Map

The locations not used (between H'000 and H'2F2) are reserved and cannot be accessed.

21.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit / receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. The following table shows the address map for the control, LAFM, data and addresses for each mailbox.

	Address			
	Control0	**LAFM**	**Data**	**Control1**
Mailbox	**4 bytes**	**4 bytes**	**8 bytes**	**2 bytes**
0 (Receive Only)	100 – 103	104– 107	108 – 10F	110 – 111
1	120 – 123	124 – 127	128 – 12F	130 – 131
2	140 – 143	144 – 147	148 – 14F	150 – 151
3	160 – 163	164 - 167	168 – 16F	170 – 171
4	180 – 183	184 – 187	188 – 18F	190 – 191
5	1A0 – 1A3	1A4 – 1A7	1A8 – 1AF	1B0 – 1B1
6	1C0 – 1C3	1C4 – 1C7	1C8 – 1CF	1D0 – 1D1
7	1E0 – 1E3	1E4 – 1E7	1E8 – 1EF	1F0 – 1F1
8	200 – 203	204 – 207	208 – 20F	210 – 211
9	220 – 223	224 – 227	228 – 22F	230 – 231
10	240 – 243	244 – 247	248 – 24F	250 – 251
11	260 – 263	264 – 267	268 – 26F	270 – 271
12	280 – 283	284 – 287	288 – 28F	290 – 291
13	2A0 – 2A3	2A4 – 2A7	2A8 – 2AF	2B0 – 2B1
14	2C0 – 2C3	2C4 – 2C7	2C8 – 2CF	2D0 – 2D1
15	2E0 – 2E3	2E4 – 2E7	2E8 – 2EF	2F0 – 2F1

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

Table 21.1 Roles of Mailboxes

	Tx	Rx
MB15-1	OK	OK
MB0	—	OK

MB0 (reception MB)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord): 32-bit access

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

MBC[1] is fixed to "1"

MB15-1 (MB for transmission/reception)

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + N*32	EXTID[15:0]																Word	
H'104 + N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
H'106 + N*32	EXTID_LAFM[15:0]																Word	
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
H'110 + N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

Figure 21.3 Mailbox-N Structure

Notes: 1. All bits shadowed in grey are reserved and must be written LOW. The value returned by a read may not always be '0' and should not be relied upon.

2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.

3. ID Reorder (MCR15) can change the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

(1) Message Control Field

STDID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXTID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit) : Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC=001(bin), the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt), however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: In order to support automatic answer to remote frame when MBC=001(bin) is used and ATX=1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit) : Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

- Mailbox-0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Note: MBC[1] of MB0 is always "1".

- Mailbox-15 to 1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

NMC (New Message Control): When this bit is set to ‘0’, the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to ‘1’, the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

Important: Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

NMC	Description
0	Overrun mode (Initial value)
1	Overwrite mode

ATX (Automatic Transmission of Data Frame): When this bit is set to ‘1’ and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be ‘001’ (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

Important: When ATX is used and MBC=001 (Bin) the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

Important: Please note that in case of overrun condition (UMSR flag set when the Mailbox has its NMC = 0) the message received is discarded. In case a remote frame is causing overrun into a Mailbox configured with ATX = 1, the transmission of the corresponding data frame may be triggered only if the related RFPR flag is cleared by the CPU when the UMSR flag is set. In such case RFPR flag would get set again.

ATX	Description
0	Automatic Transmission of Data Frame disabled (Initial value)
1	Automatic Transmission of Data Frame enabled

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to ‘0’, RCAN-ET tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

DART	Description
0	Re-transmission enabled (Initial value)
1	Re-Transmission disabled

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC=111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC=’110’, ‘101’ and ‘100’ settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception. There is no hardware protection, and TXPR remains set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX* • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	Setting prohibited				
1	0	1	Setting prohibited				
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive (Initial value)				

Notes: * In order to support automatic retransmission, RTR shall be "0" when MBC=001(bin) and ATX=1.
When ATX=1 is used the filter for IDE must not be used

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 bytes (Initial value)
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

(2) Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes**.**

LAFM: When MBC is set to 001, 010, 011 (Bin), this field is used as LAFM Field. It allows a Mailbox to accept more than one identifier. The LAFM is comprised of two 16-bit read/write areas as follows.

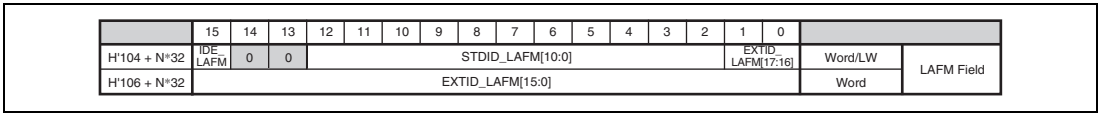


Figure 21.4 Acceptance Filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-ET searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

Important: RCAN-ET starts to find a matching identifier from Mailbox-15 down to Mailbox-0. As soon as RCAN-ET finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

STD_LAFM[10:0] — Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding STD_ID bit is cared
1	Corresponding STD_ID bit is "don't cared"

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE_ID bit is cared
1	Corresponding IDE_ID bit is "don't cared"

(3) Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

21.3.3 RCAN-ET Control Registers

The following sections describe RCAN-ET control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Description	Address	Name	Access Size (bits)
Master Control Register	000	MCR	Word
General Status Register	002	GSR	Word
Bit Configuration Register 1	004	BCR1	Word
Bit Configuration Register 0	006	BCR0	Word
Interrupt Request Register	008	IRR	Word
Interrupt Mask Register	00A	IMR	Word
Error Counter Register	00C	TEC/REC	Word

Figure 21.5 RCAN-ET Control Registers

(1) Master Control Register (MCR)

The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-ET.

- MCR (Address = H'000)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	-	-	-	TST[2:0]			MCR7	MCR6	MCR5	-	-	MCR2	MCR1	MCR0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit 15 — ID Reorder (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

Bit15 : MCR15	Description
0	RCAN-ET is the same as HCAN2
1	RCAN-ET is not the same as HCAN2 (Initial value)

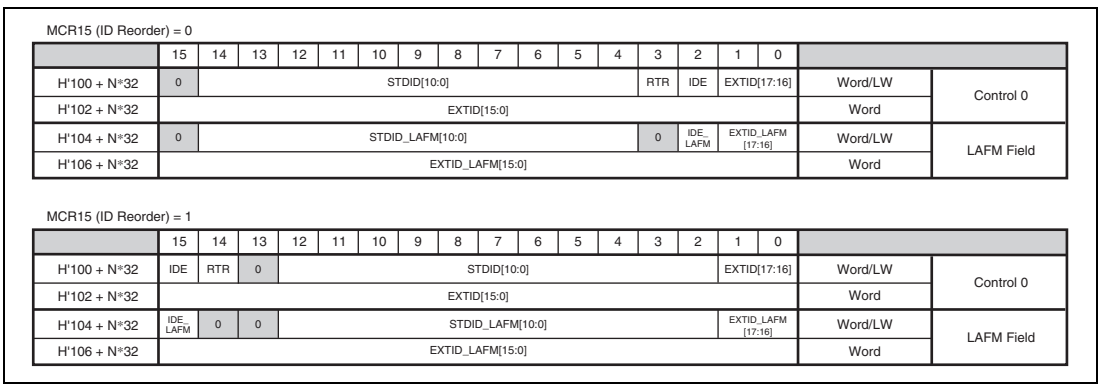


Figure 21.6 ID Reorder

This bit can be modified only in reset mode.

Bit 14 — Auto Halt Bus Off (MCR14): If both this bit and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

Bit14 : MCR14	Description
0	RCAN-ET remains in BusOff for normal recovery sequence (128 x 11 Recessive Bits) (Initial value)
1	RCAN-ET moves directly into Halt Mode after it enters BusOff if MCR6 is set.

This bit can be modified only in reset mode.

Bit 13 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 12 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 11 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 10 - 8 — Test Mode (TST[2:0]): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move RCAN-ET into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 21.4.1, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-ET is used in normal operation.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	setting prohibited
1	1	1	setting prohibited

Bit 7 — Auto-wake Mode (MCR7): MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-ET automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-ET does not automatically cancel the sleep mode.

RCAN-ET cannot store the message that wakes it up.

Note: MCR7 cannot be modified while in sleep mode.

Bit7 : MCR7	Description
0	Auto-wake by CAN bus activity disabled (Initial value)
1	Auto-wake by CAN bus activity enabled

Bit 6 — Halt during Bus Off (MCR6): MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

Bit6 : MCR6	Description
0	If MCR[1] is set, RCAN-ET will not enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value)
1	Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.

Bit 5 — Sleep Mode (MCR5): Enables or disables Sleep mode transition. If this bit is set, while RCAN-ET is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position,
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-ET will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-ET will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-ET will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

Important: RCAN-ET is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-ET must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5]=1 and MCR[1]=0 at the same time).

Bit 5 : MCR5	Description
0	RCAN-ET sleep mode released (Initial value)
1	Transition to RCAN-ET sleep mode enabled

Bit 4 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 2 — Message Transmission Priority (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-15 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission).

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE=1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

Bit 2 : MCR2	Description
0	Transmission order determined by message identifier priority (Initial value)
1	Transmission order determined by mailbox number priority (Mailbox-15 → Mailbox-1)

Bit 1—Halt Request (MCR1): Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-ET remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-ET will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

In the Halt mode, the RCAN-ET configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a ‘0’ in order to re-join the CAN bus. After this bit has been cleared, RCAN-ET waits until it detects 11 recessive bits, and then joins the CAN bus.

Note: After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).

Note: Transition into or recovery from HALT mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

Bit 1 : MCR1	Description
0	Clear Halt request (Initial value)
1	Halt mode transition request

Bit 0 — Reset Request (MCR0): Controls resetting of the RCAN-ET module. When this bit is changed from '0' to '1' the RCAN-ET controller enters its reset routine, re-initialising the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all user registers are initialised.

RCAN-ET can be re-configured while this bit is set. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, the RCAN-ET module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-ET needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

Bit 0 : MCR0	Description
0	Clear Reset Request
1	CAN Interface reset mode transition request (Initial value)

(2) General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-ET.

- GSR (Address = H'002)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-ET enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

Bit 5 : GSR5	Description
0	RCAN-ET is not in Error Passive or in Bus Off status (Initial value) [Reset condition] RCAN-ET is in Error Active state
1	RCAN-ET is in Error Passive (if GSR0=0) or Bus Off (if GSR0=1) [Setting condition] When TEC ≥ 128 or REC ≥ 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN-ET IP. RCAN-ET exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4 : GSR4	Description
0	RCAN-ET is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1=1) or Sleep mode (if MCR5=1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and RCAN-ET is in the halt mode or RCAN-ET is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-ET is in the reset state or not.

Bit 3 : GSR3	Description
0	RCAN-ET is not in the reset state
1	Reset state (Initial value) [Setting condition] After an RCAN-ET internal reset (due to SW or HW reset)

Bit 2 — Message Transmission in progress Flag (GSR2): Flag that indicates to the CPU if the RCAN-ET is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

Bit 2 : GSR2	Description
0	RCAN-ET is in Bus Off or a transmission is in progress
1	[Setting condition] Not in Bus Off and no transmission in progress (Initial value)

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

Bit 1 : GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC < 256 or 96 ≤ REC < 256

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that RCAN-ET is in the bus off state.

Bit 0 : GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC ≥ 256 (bus off state)

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is equivalent to GSR0.

(3) Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 X 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$Timequanta = \frac{2 * BRP}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral bus frequency.

- BCR1 (Address = H'004)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				-	TSG2[2:0]			-	-	SJW[1:0]		-	-	-	BSP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Bits 15 to 12 — Time Segment 1 (TSG1[3:0] = BCR1[15:12]): These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: TSG1[3]	Bit 14: TSG1[2]	Bit 13: TSG1[1]	Bit 12: TSG1[0]	Description
0	0	0	0	Setting prohibited (Initial value)
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	PRSEG + PHSEG1 = 4 time quanta
0	1	0	0	PRSEG + PHSEG1 = 5 time quanta
:	:	:	:	:
:	:	:	:	:
1	1	1	1	PRSEG + PHSEG1 = 16 time quanta

Bit 11 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 10 to 8 — Time Segment 2 (TSG2[2:0] = BCR1[10:8]): These bits are used to set the segment TSEG2 (=PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

Bit 10: TSG2[2]	Bit 9: TSG2[1]	Bit 8: TSG2[0]	Description
0	0	0	Setting prohibited (Initial value)
0	0	1	PHSEG2 = 2 time quanta (conditionally prohibited)
0	1	0	PHSEG2 = 3 time quanta
0	1	1	PHSEG2 = 4 time quanta
1	0	0	PHSEG2 = 5 time quanta
1	0	1	PHSEG2 = 6 time quanta
1	1	0	PHSEG2 = 7 time quanta
1	1	1	PHSEG2 = 8 time quanta

Bits 7 and 6 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

Bit 5: SJW[1]	Bit 4: SJW[0]	Description
0	0	Synchronisation Jump width = 1 time quantum (Initial value)
0	1	Synchronisation Jump width = 2 time quanta
1	0	Synchronisation Jump width = 3 time quanta
1	1	Synchronisation Jump width = 4 time quanta

Bits 3 to 1 : Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled.

Bit 0 : BSP	Description
0	Bit sampling at one point (end of time segment 1) (Initial value)
1	Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1)

- BCR0 (Address = H'006)

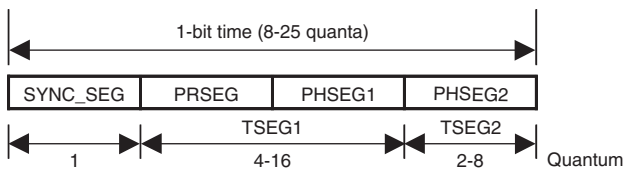
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 8 to 15 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral bus clock periods contained in a Time Quantum.

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 × peripheral bus clock (Initial value)
0	0	0	0	0	0	0	1	4 × peripheral bus clock
0	0	0	0	0	0	1	0	6 × peripheral bus clock
: :	: :	: :	: :	: :	: :	: :	: :	2 × (register value+1) × peripheral bus clock
0	1	1	1	1	1	1	1	512 × peripheral bus clock

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation or resynchronisation is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronisation or resynchronisation is established)

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

BRP: BRP[7:0] (bits 7 to 0 in BCR0)

The RCAN-ET Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 * (BRP + 1) * (TSEG1 + TSEG2 + 1)}$$

where BRP is given by the register value, and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values. The '+ 1' in the above formula is for the Sync-Seg which duration is 1 time quanta.

f_{CLK} = Peripheral Clock

BCR Setting Constraints

TSEG1min > TSEG2 ≥ SJWmax (SJW = 1 to 4)

8 ≤ TSEG1 + TSEG2 + 1 ≤ 25 time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

TSEG2 ≥ 2

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	No	1-3	No	No	No	No	No	
0100	5	1-2	1-3	1-4	No	No	No	No	
0101	6	1-2	1-3	1-4	1-4	No	No	No	
0110	7	1-2	1-3	1-4	1-4	1-4	No	No	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	No	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

Example 1: To have a Bit rate of 500 Kbps with a frequency of fclk = 40 MHz it is possible to set: BRP = 3, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = 5200 and BCR0 = 0003.

Example 2: To have a Bit rate of 250 Kps with a frequency of 35 MHz it is possible to set: BPR = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

(4) Interrupt Request Register (IRR)

The interrupt register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	IRR13	IRR12	-	-	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bits 15 to 14: Reserved.

Bit 13 - Message Error Interrupt (IRR13): this interrupt indicates that:

- A message error has occurred when in test mode.
- Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set. When not in test mode this interrupt is inactive.

Bit 13: IRR13	Description
0	message error has not occurred in test mode (Initial value) [Clearing condition] Writing 1
1	[Setting condition] message error has occurred in test mode

Bit 12 – Bus activity while in sleep mode (IRR12): IRR12 indicates that a CAN bus activity is present. While the RCAN-ET is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

Bit 12: IRR12	Description
0	bus idle state (Initial value) [Clearing condition] Writing 1
1	[Setting condition] dominant bit level detection on the Rx line while in sleep mode

Bits 11 to 10: Reserved

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to ‘1’ and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing ‘1’) or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set . It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 9: IRR9	Description
0	No pending notification of message overrun/overwrite [Clearing condition] Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)
1	A receive message has been discarded due to overrun condition or a message has been overwritten [Setting condition] Message is received while the corresponding RXPR and/or RFPR =1 and MBIMR =0

Bit 8 - Mailbox Empty Interrupt Flag (IRR8): This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). The related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In effect, this bit is set by an OR’ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 8: IRR8	Description
0	Messages set for transmission or transmission cancellation request NOT progressed. (Initial value) [Clearing Condition] All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set
1	Message has been transmitted or aborted, and new message can be stored [Setting condition] When one of the TXPR bits is cleared by completion of transmission or completion of transmission abort, i.e., when a TXACK or ABACK bit is set (if MBIMR=0).

Bit 7 — Overload Frame (IRR7): Flag indicating that the RCAN-ET has detected a condition that should initiate the transmission of an overload frame. Note that on the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

Bit 7: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	[Setting conditions] Overload condition detected

Bit 6 — Bus Off Interrupt Flag (IRR6): This bit is set when RCAN-ET enters the Bus-off state or when RCAN-ET leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128X11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-ET node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-ET is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

Bit 6: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Enter Bus off state caused by transmit error or Error Active state returning from Bus-off [Setting condition] When TEC becomes ≥ 256 or End of Bus-off after 128X11 consecutive recessive bits or transition from Bus Off to Halt

Bit 5 — Error Passive Interrupt Flag (IRR5): Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-ET is in Error Passive or Bus Off status.

Bit 5: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128 or Error Passive test mode is used

Bit 4 — Receive Error Counter Warning Interrupt Flag (IRR4): This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when RCAN-ET is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 4: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96 and RCAN-ET is not in Bus Off

Bit 3 — Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 3: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 2 — Remote Frame Request Interrupt Flag (IRR2): flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 2: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (Initial value)
1	at least one remote request is pending [Setting condition] When remote frame is received and the corresponding MBIMR = 0

Bit 1 — Data Frame Received Interrupt Flag (IRR1): IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 1: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (Initial value)
1	Data frame received and stored in Mailbox [Setting condition] When data is received and the corresponding MBIMR = 0

Bit 0 — Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

1. Reset mode has been entered after a SW (MCR0) or HW reset
2. Halt mode has been entered after a Halt request (MCR1)
3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state RCAN-ET is in.

Important : When a Sleep mode request needs to be made, the Halt mode must be used beforehand. Please refer to the MCR5 description and figure 21.9.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-ET enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time * 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialisation.

Bit 0: IRR0	Description
0	[Clearing condition] Writing 1
1	Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value) [Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

(5) Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0: Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

Bit[15:0]: IMRn	Description
0	Corresponding IRR is not masked (IRQ is generated for interrupt conditions)
1	Corresponding interrupt of IRR is masked (Initial value)

(6) Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [1], [2], [3] and [4]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = 3'b100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-ET needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is only possible to write the value in test mode when TST[2:0] in MCR is 3'b100.
REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

21.3.4 RCAN-ET Mailbox Registers

The following sections describe RCAN-ET Mailbox registers that control / flag individual Mailboxes. The address is mapped as follows.

Important : LongWord access is carried out as two consecutive Word accesses.

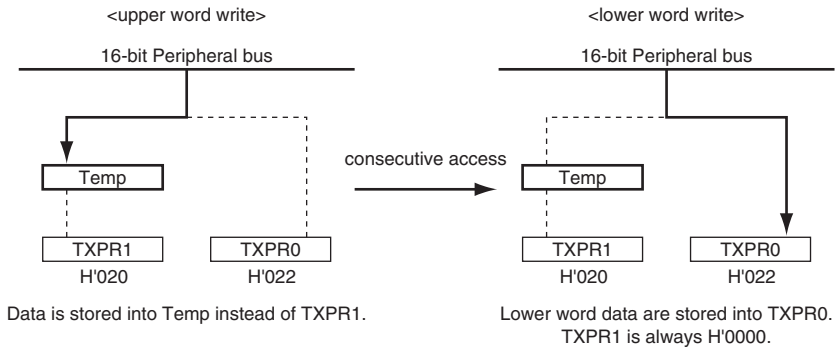
Description	Address	Name	Access Size (bits)
Transmit Pending 1	H'020	TXPR1	LW
Transmit Pending 0	H'022	TXPR0	—
	H'024		
	H'026		
	H'028		
Transmit Cancel 0	H'02A	TXCR0	
	H'02C		
	H'02E		
	H'030		
Transmit Acknowledge 0	H'032	TXACK0	Word
	H'034		
	H'036		
	H'038		
Abort Acknowledge 0	H'03A	ABACK0	Word
	H'03C		
	H'03E		
	H'040		
Data Frame Receive Pending 0	H'042	RXPR0	Word
	H'044		
	H'046		
	H'048		
Remote Frame Receive Pending 0	H'04A	RFPR0	Word
	H'04C		
	H'04E		
	H'050		
Mailbox Interrupt Mask Register 0	H'052	MBIMR0	Word
	H'054		
	H'056		
	H'058		
Unread message Status Register 0	H'05A	UMSR0	Word
	H'05C		
	H'05E		

Figure 21.7 RCAN-ET Mailbox Registers

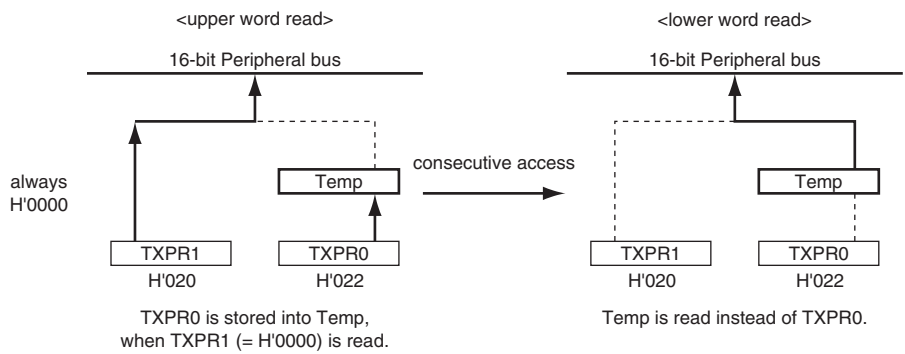
(1) Transmit Pending Register (TXPR1, TXPR0)

The concatenation of TXPR1 and TXPR0 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

<Longword Write Operation>



<Longword Read Operation>



The TXPR1 register cannot be modified and it is always fixed to ‘0’. The TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a ‘0’ and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

The RCAN-ET will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-ET automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-ET shall ensure that in the identifier priority scheme (MCR2=0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to section 21.4, Application Note.

When the RCAN-ET changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Any write operation is ignored.

Read value is always H'0000. Long word access is mandatory when reading or writing TXPR1/TXPR0. Writing any value to TXPR1 is allowed, however, write operation to TXPR1 has no effect.

- TXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * it is possible only to write a ‘1’ for a Mailbox configured as transmitter.

Bit 15 to 1 — indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:1]:TXPR0	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

Bit 0— Reserved: This bit is always ‘0’ as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

(2) Transmit Cancel Register (TXCR0)

TXCR0 is a 16-bit read / conditionally-write registers. The TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * Only writing a ‘1’ to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 1 — requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

Bit[15:1]:TXCR0	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(3) Transmit Acknowledge Register (TXACK0)

The TXACK0 is a 16-bit read / conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-ET sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 1 — notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:TXACK0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(4) Abort Acknowledge Register (ABACK0)

The ABACK0 is a 16-bit read / conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-ET sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-ET to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	-

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 1 — notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:ABACK0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

Bit 0 — This bit is always ‘0’ as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a ‘0’.

(5) Data Frame Receive Pending Register (RXPR0)

The RXPR0 is a 16-bit read / conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

Bit[15:0]: RXPR0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

(6) Remote Frame Receive Pending Register (RFPR0)

The RFPR0 is a 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Request Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note : * Only when writing a ‘1’ to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 15 to 0 respectively.

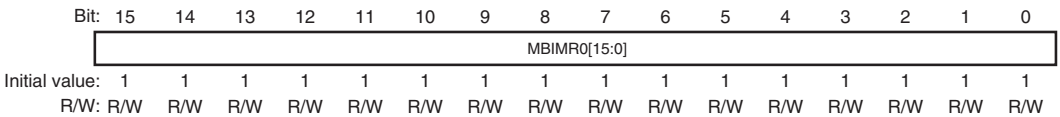
Bit[15:0]: RFPR0	Description
0	[Clearing Condition] Writing ‘1’ (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

(7) Mailbox Interrupt Mask Register (MBIMR)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Request Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR0



Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

Bit[15:0]: MBIMR0	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

(8) Unread Message Status Register (UMSR)

This register is a 16-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

Bit[15:0]: UMSR0	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

21.4 Application Note

21.4.1 Test Mode Settings

The RCAN-ET has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-ET test mode. The default (initialised) settings allow RCAN-ET to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal Mode (initial value)
0	0	1	Listen-Only Mode (Receive-Only Mode)
0	1	0	Self Test Mode 1 (External)
0	1	1	Self Test Mode 2 (Internal)
1	0	0	Write Error Counter
1	0	1	Error Passive Mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Normal Mode: RCAN-ET operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the Tx Output is disabled so that RCAN-ET does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

Self Test Mode 1: RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The Rx/Tx pins must be connected to the CAN bus.

Self Test Mode 2: RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The Rx/Tx pins do not need to be connected to the CAN bus or any external devices, as the internal Tx is looped back to the internal Rx. Tx pin outputs only recessive bits and Rx pin is disabled.

Write Error Counter: TEC/REC can be written in this mode. RCAN-ET can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-ET can be forced to become an Error Warning by writing a value greater than 95 into them.

RCAN-ET needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.Error Passive Mode: RCAN-ET can be forced to enter Error Passive mode.

Note: the REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, RCAN-ET will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-ET will not be able to become Error Active. Consequently, at the end of the Bus Off recovery sequence, RCAN-ET will move to Error Passive and not to Error Active

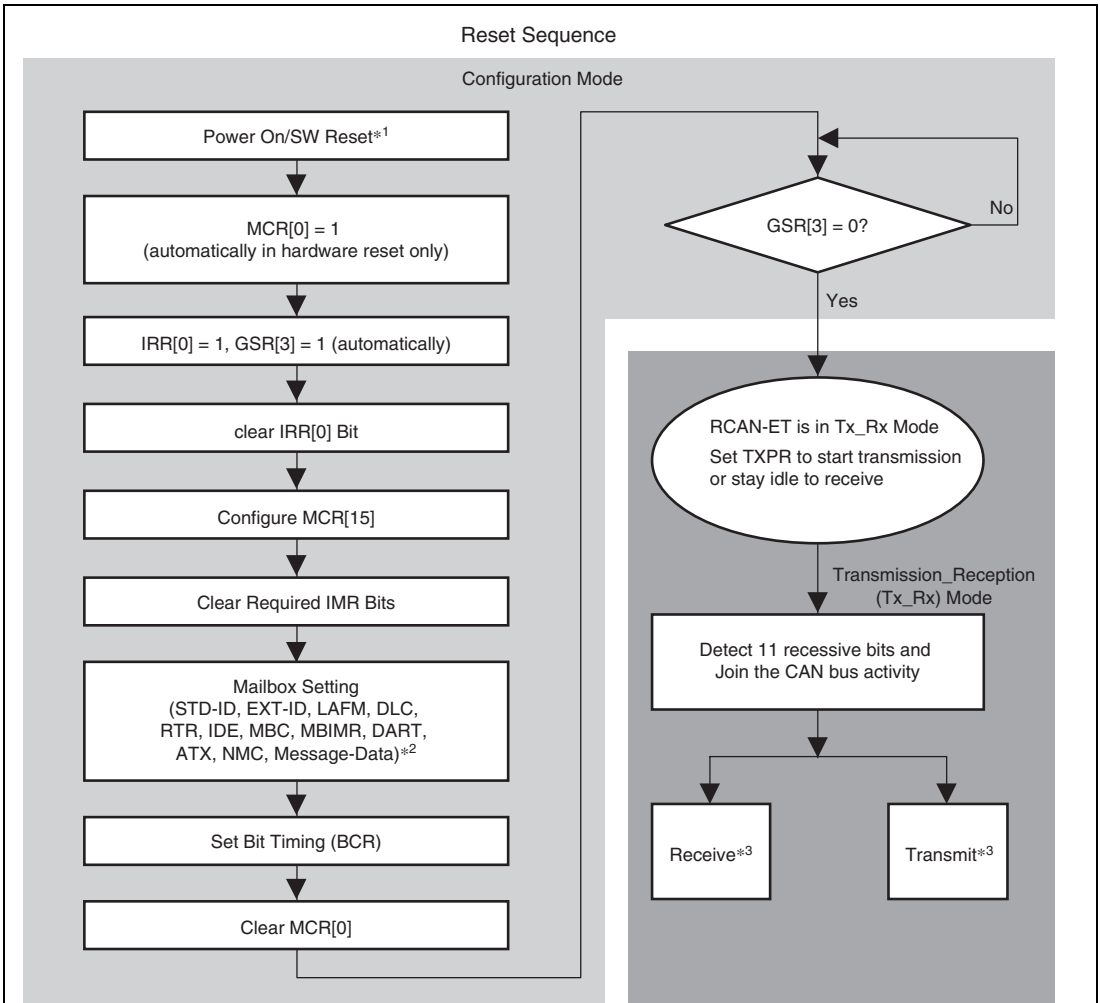
When message error occurs, IRR13 is set in all test modes.

21.4.2 Configuration of RCAN-ET

RCAN-ET is considered in configuration mode or after a H/W (Power On Reset)/ S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-ET cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

- After a Reset request

  The following sequence must be implemented to configure the RCAN-ET after (S/W or H/W) reset. After reset, all the registers are initialised, therefore, RCAN-ET needs to be configured before joining the CAN bus activity. Please read the notes carefully.



Notes: 1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialise all the mailboxes enabled by MBC.
3. If there is no TXPR set, RCAN-ET will receive the next incoming message.
If there is a TXPR(s) set, RCAN-ET will start transmission of the message and will be arbitrated by the CAN bus.
If it loses the arbitration, it will become a receiver.

Figure 21.8 Reset Sequence

- Halt mode

  When RCAN-ET is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-ET to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-ET transit to Halt Mode, GSR4 is set.

  Once the configuration is completed the Halt request needs to be released. RCAN-ET will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

- Sleep mode

  When RCAN-ET is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

  The following diagram shows the flow to follow to move RCAN-ET into sleep mode.

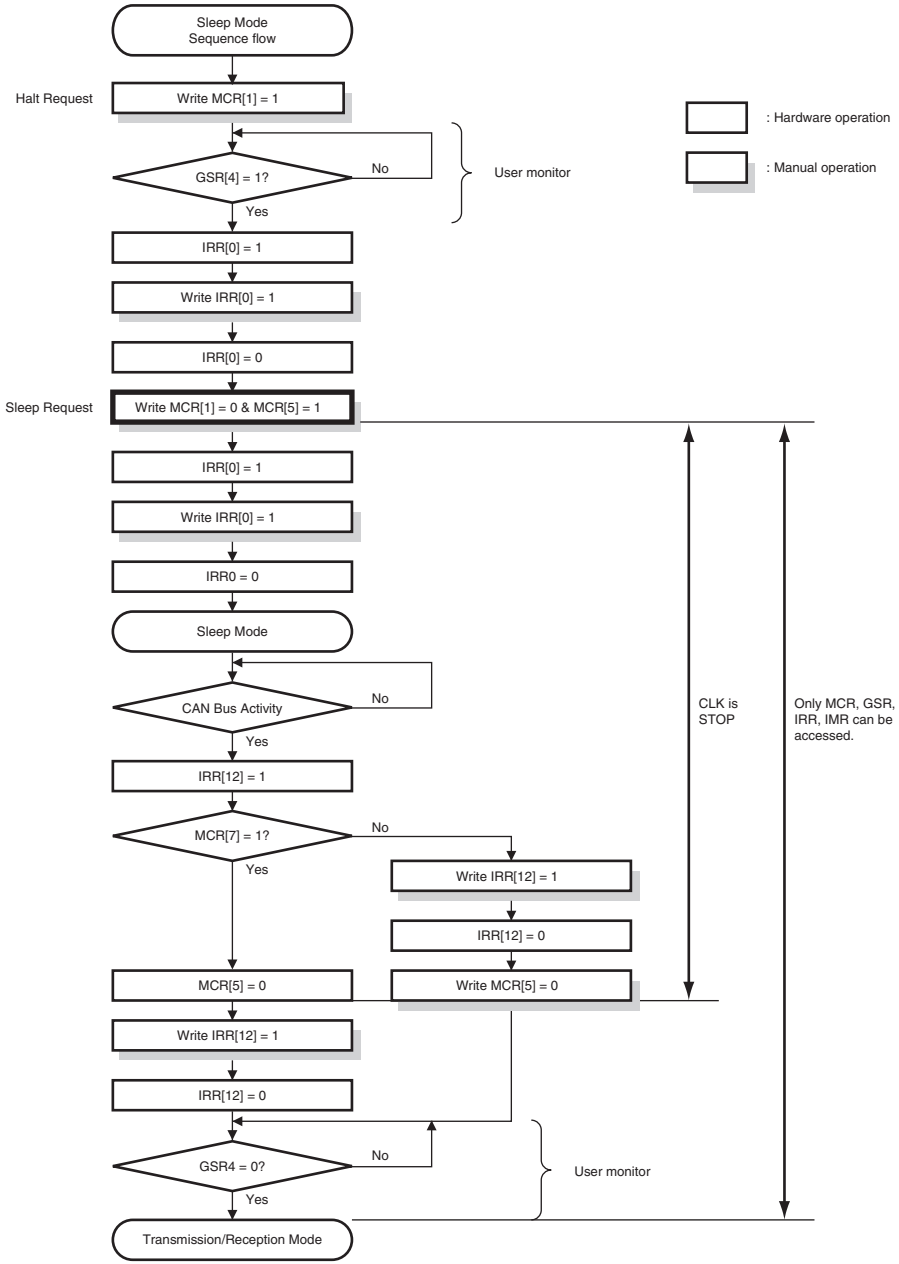

Sleep Mode Sequence flow
Halt Request
Write MCR[1] = 1
GSR[4] = 1?
No
Yes
User monitor
: Hardware operation
: Manual operation
IRR[0] = 1
Write IRR[0] = 1
IRR[0] = 0
Sleep Request
Write MCR[1] = 0 & MCR[5] = 1
IRR[0] = 1
Write IRR[0] = 1
IRR0 = 0
Sleep Mode
CAN Bus Activity
No
Yes
CLK is STOP
Only MCR, GSR, IRR, IMR can be accessed.
IRR[12] = 1
MCR[7] = 1?
No
Yes
Write IRR[12] = 1
IRR[12] = 0
MCR[5] = 0
Write MCR[5] = 0
Write IRR[12] = 1
IRR[12] = 0
GSR4 = 0?
No
Yes
User monitor
Transmission/Reception Mode

Figure 21.9 - Halt Mode / Sleep Mode shows allowed state transition.

— Please don't set MCR5 (Sleep Mode) without entering Halt Mode.
— After MCR1 is set, please don't clear it before GSR4 is set and RCAN-ET enters Halt Mode.

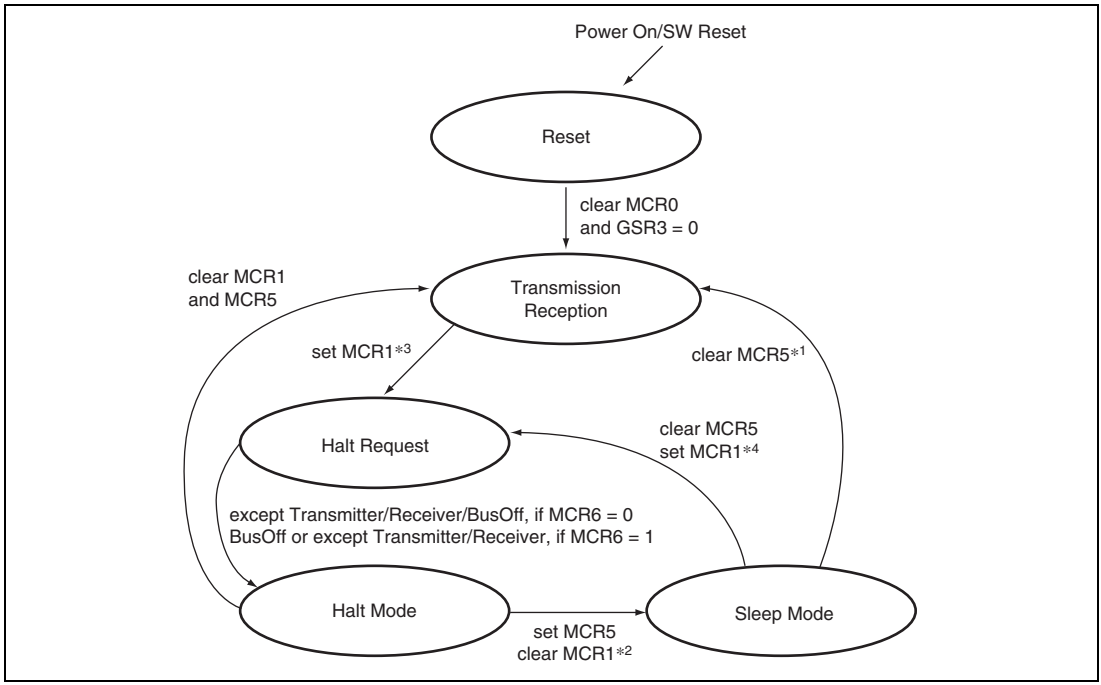


Figure 21.9 Halt Mode / Sleep Mode

Notes: 1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing "0"
2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.
3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-ET moves to Bus Off and MCR14 and MCR6 are both set.
4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-ET moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

The following table shows conditions to access registers.

	RCAN-ET Registers								
Status Mode	**MCR GSR**	**IRR IMR**	**BCR**	**MBIMR**	**Flag_register**	**mailbox (ctrl0, LAFM)**		**mailbox (data)**	**mailbox (ctrl1)**	
Reset	yes	yes	yes	yes	yes	yes		yes	yes	
Transmission Reception Halt Request	yes	yes	no*1	yes	yes	no*1	yes*2	yes*2	no*1	yes*2
Halt	yes	yes	no*1	yes	yes	yes		yes	yes	
Sleep	yes	yes	no	no	no	no		no	no	

Notes: 1. No hardware protection
2. When TXPR is not set.

21.4.3 Message Transmission Sequence

- Message Transmission Request

  The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

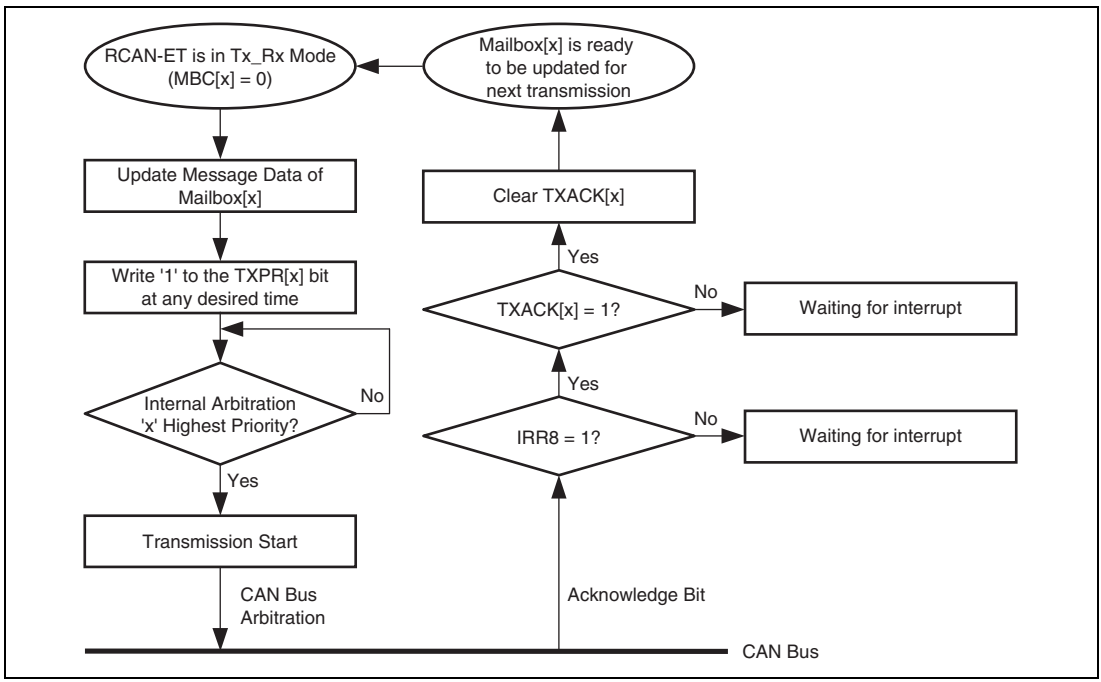


Figure 21.10 Transmission Request

- Internal Arbitration for transmission

  The following diagram explains how RCAN-ET manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

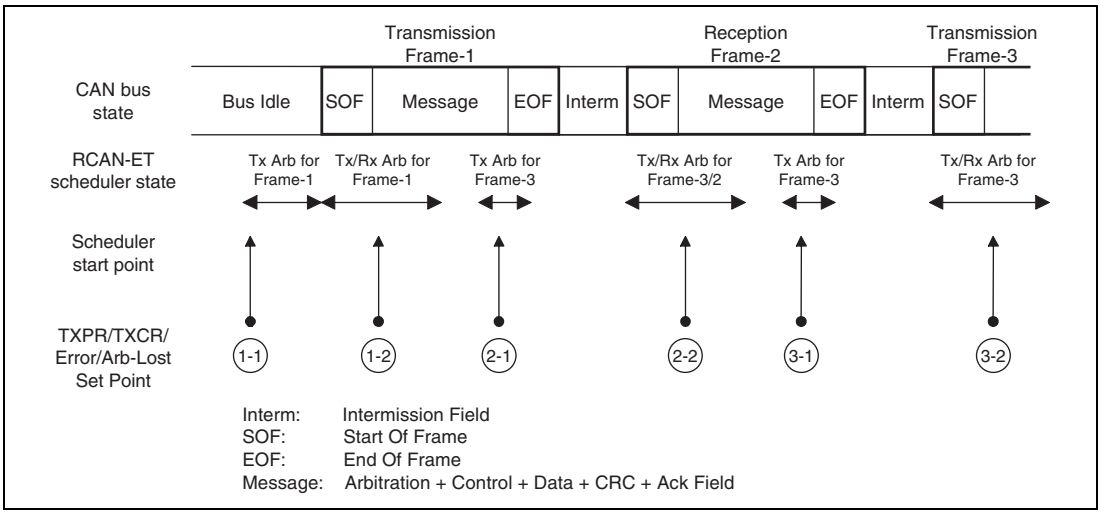


Figure 21.11 Internal Arbitration for Transmission

The RCAN-ET has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-ET becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-ET becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-ET becomes transmitter.

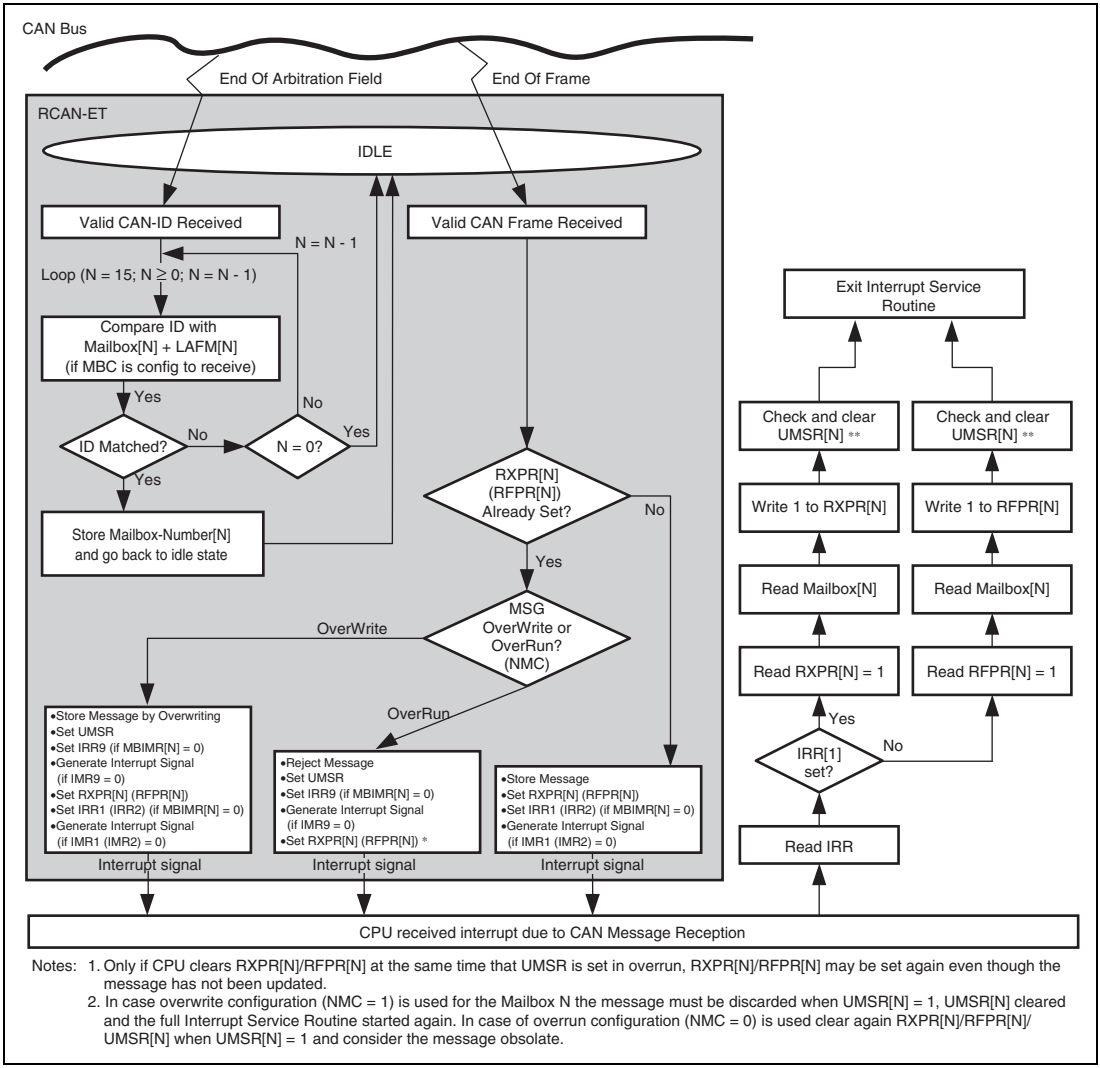
Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX=1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

21.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Notes: 1. Only if CPU clears RXPR[N]/RFPR[N] at the same time that UMSR is set in overrun, RXPR[N]/RFPR[N] may be set again even though the message has not been updated.
2. In case overwrite configuration (NMC = 1) is used for the Mailbox N the message must be discarded when UMSR[N] = 1, UMSR[N] cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/RFPR[N]/ UMSR[N] when UMSR[N] = 1 and consider the message obsolate.

Figure 21.12 Message Receive Sequence

When RCAN-ET recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-15 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-15 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-14 (if configured as receive). Once RCAN-ET finds a matching identifier, it stores the number of Mailbox-[N] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

In case UMSR is set and the Mailbox is configured for overrun (NMC = 0) the message is still valid, however it is obsolete as it is not reflecting the latest message monitored on the CAN Bus. Please access the full Mailbox content before clearing the related RXPR/RFPR flag.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame request interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

In the Overrun Mode (NMC = ’0’), only the first Mailbox will cause the flags to be asserted. So, if a Data Frame is initially received, then RXPR and IRR1 are both asserted. If a Remote Frame is then received before the Data Frame has been read, then RFPR and IRR2 are NOT set. In this case UMSR of the corresponding Mailbox will still be set.

21.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

- Change configuration of transmit box

  Two cases are possible.

  — Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART

  This change is possible only when MBC=3'b000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.

  — Change from transmit to receive configuration (MBC)

  Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-ET to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt state.

  In case RCAN-ET is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

- Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART, MBC) of receiver box or Change receiver box to transmitter box

  The configuration can be changed only in Halt Mode.

  RCAN-ET will not lose a message if the message is currently on the CAN bus and RCAN-ET is a receiver. RCAN-ET will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-ET is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt Mode.

  In case RCAN-ET is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

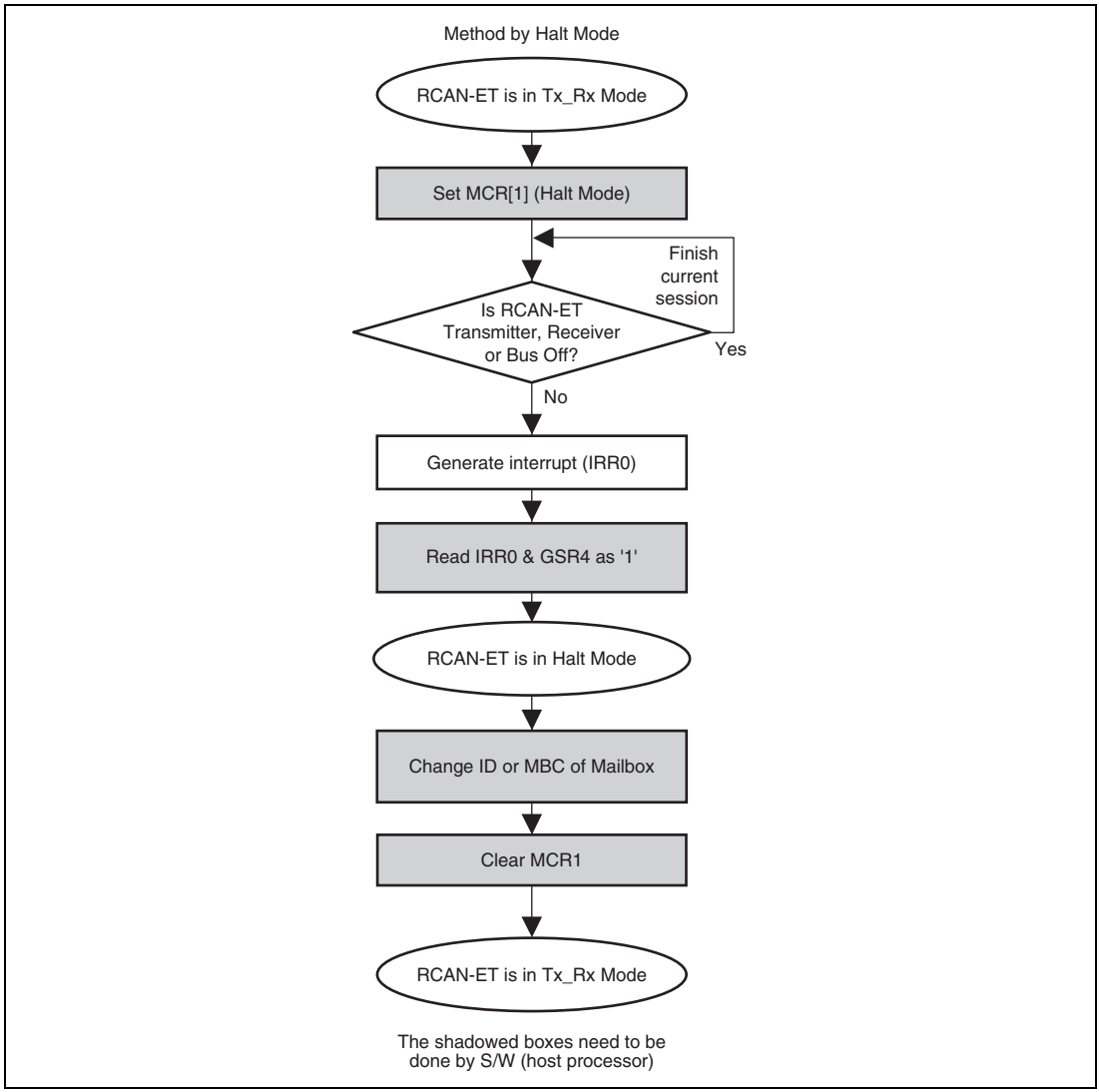


Figure 21.13 Change ID of Receive Box or Change Receive Box to Transmit Box

21.5 Interrupt Sources

Table 21.2 lists the RCAN-ET interrupt sources. With the exception of the reset processing interrupt (IRR0) by a power-on reset, these sources can be masked. Masking is implemented using the mailbox interrupt mask register 0 (MBIMR0) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 6, Interrupt Controller (INTC).

Table 21.2 RCAN-ET Interrupt Sources

Module	Interrupt	Description	Interrupt Flag	DTC Activation
RCAN-ET	ERS_0	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
		Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
		Error warning (TEC ≥ 96)	IRR3	
		Error warning (REC ≥ 96)	IRR4	
	OVR_0	Message error detection	IRR13*1	
		Reset/halt/CAN sleep transition	IRR0	
		Overload frame transmission	IRR7	
		Unread message overwrite (overrun)	IRR9	
		Detection of CAN bus operation in CAN sleep mode	IRR12	
	RM0_0*2	Data frame reception	IRR1*3	Possible*4
	RM1_0*2	Remote frame reception	IRR2*3	
	SLE_0	Message transmission/transmission disabled (slot empty)	IRR8	Not possible

Notes: 1. Available only in Test Mode.
2. RM0_0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1_0 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 15).
3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 15, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 15.
4. The DTC can be activated only by the RM0_0 interrupt.

21.6 DTC Interface

The DTC can be activated by the reception of a message in RCAN-ET mailbox 0. When DTC transfer ends after DTC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-ET cannot be sent to the CPU in this case. Figure 21.14 shows a DTC transfer flowchart.

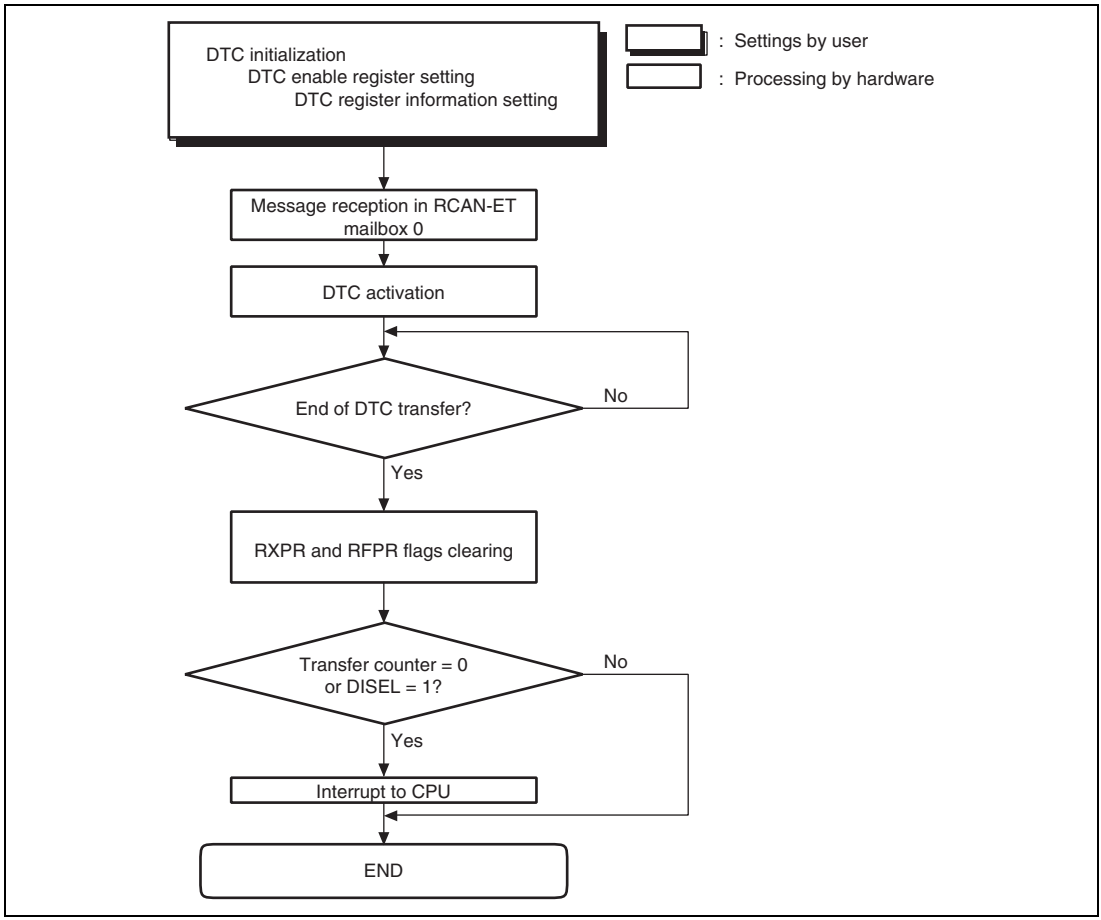
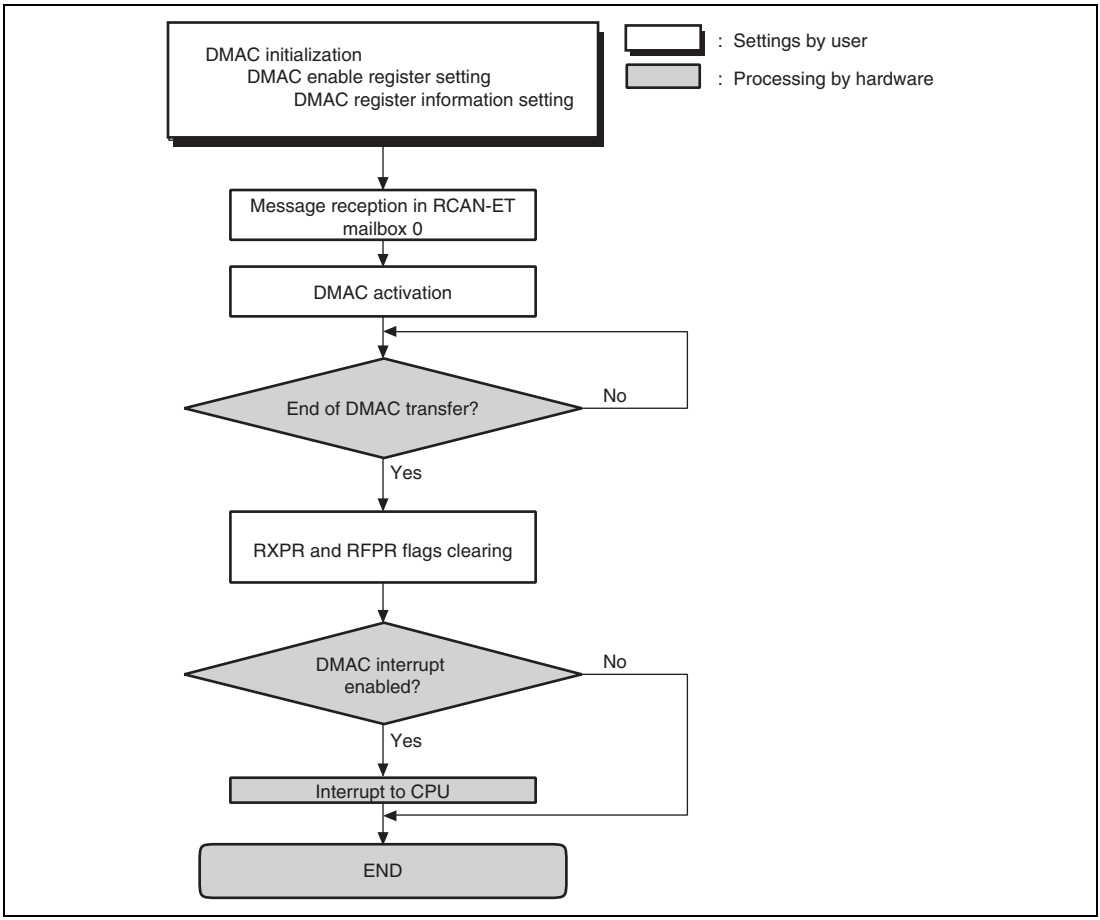


Figure 21.14 DTC Transfer Flowchart

21.7 DMAC Interface

The DMAC can be activated by the reception of a message in RCAN-ET mailbox 0. When DMAC transfer ends after DMAC activation has been set, flags of RXPR0 and RFPR0 are cleared automatically. An interrupt request due to a receive interrupt from the RCAN-ET cannot be sent to the CPU in this case. Figure 21.15 shows a DMAC transfer flowchart.



21.15 DMAC Transfer Flowchart

21.8 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. The specification for this LSI circuit is a 3-V power-supply voltage, so use a level-shifter IC between its CRx0 pin and the Rxd pin of the HA13721. Figure 21.16 shows a sample connection diagram.

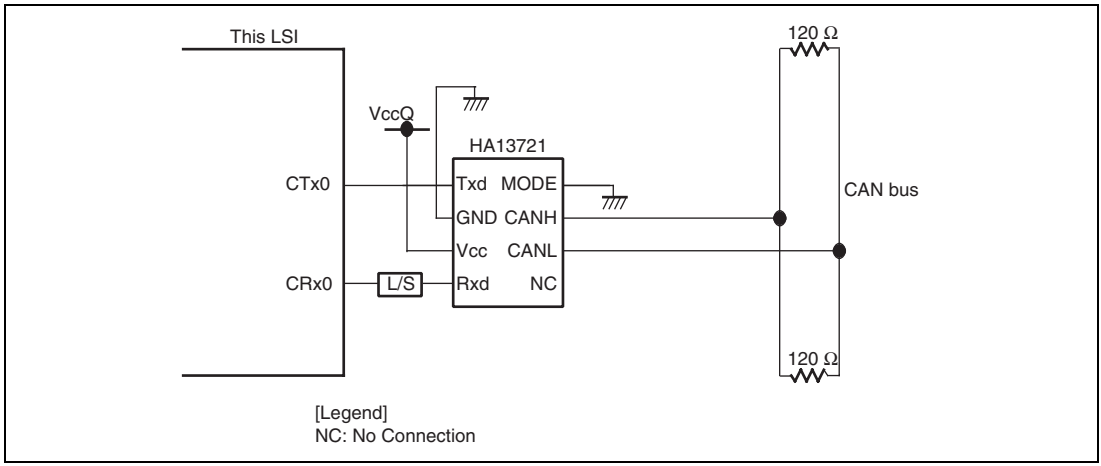


Figure 21.16 High-Speed CAN Interface Using HA13721

21.9 Usage Notes

21.9.1 Module Standby Mode

The clock supply to RCAN-ET can be stopped or started by using the standby control register 6 (STBCR6). With the initial value, the clock supply is stopped. Access to the RCAN-ET registers should be made only after releasing RCAN-ET from module standby mode.

21.9.2 Reset

RCAN-ET can be reset by hardware reset or software reset.

- Hardware reset
  RCAN-ET is reset to the initial state by power-on reset or on entering module standby mode.
- Software reset
  By setting the MCR0 bit in Master Control Register (MCR), RCAN-ET registers, excluding the MCR0 bit, and the CAN communication circuitry are initialized.

Since the IRR0 bit in Interrupt Request Register (IRR) is set by the initialization upon reset, it should be cleared while RCAN-ET is in configuration mode during the reset sequence.

The areas except for message control field 1 (CONTROL1) of mailboxes are not initialized by reset because they are in RAM. After power-on reset, all mailboxes should be initialized while RCAN-ET is in configuration mode during the reset sequence.

21.9.3 CAN Sleep Mode

In CAN sleep mode, the clock supply to the major parts in the module is stopped. Therefore, do not make access in CAN sleep mode except for access to the MCR, GSR, IRR, and IMR registers.

21.9.4 Register Access

If the mailbox area is accessed while the CAN communication circuitry in RCAN-ET is storing a received CAN bus frame in a mailbox, a 0 to five peripheral clock cycles of wait state is generated.

21.9.5 Interrupts

As shown in table 21.2, a Mailbox 0 receive interrupt can activate the DTC. If configured such that the DTC is activated by a Mailbox 0 receive interrupt and clearing of the interrupt source flag upon DTC transfer is enabled, use block transfer mode and read the whole Mailbox 0 message up to the message control field 1 (CONTROL1).

Section 22 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of registers that are used to select the functions of multiplexed pins and assign pins to be inputs or outputs. Tables 22.1 to 22.6 list the multiplexed pins of this LSI.

Table 22.1 Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
A	PA21 I/O (Port)	$\overline{RD}$ output (BSC)	$\overline{BACK}$ output (BSC)	IRQ5 input (INTC)	CKE output (BSC)	$\overline{POE3}$ input (POE2)	SCK1 I/O (SCI)	$\overline{FRAME}$ output (BSC)
	PA20 I/O (Port)	$\overline{WRL}$ output, DQMLL output (BSC)	$\overline{BREQ}$ input (BSC)	IRQ6 input (INTC)	$\overline{CASU}$ output (BSC)	$\overline{POE4}$ input (POE2)	TXD1 output (SCI)	$\overline{AH}$ output (BSC)
	PA19 I/O (Port)	$\overline{WRH}$ output, DQMLU output (BSC)	$\overline{WAIT}$ input (BSC)	IRQ7 input (INTC)	$\overline{RASU}$ output (BSC)	$\overline{POE8}$ input (POE2)	RXD1 input (SCI)	$\overline{BS}$ output (BSC)
	PA18 I/O (Port)	CK output (BSC)	—	—	—	—	—	—
	PA17 I/O (Port)	$\overline{RD}$ output (BSC)	—	—	—	—	—	—
	PA16 I/O (Port)	$\overline{WRL}$ output, DQMLL output (BSC)	—	—	—	—	—	—
	PA15 I/O (Port)	$\overline{WRH}$ output, DQMLU output (BSC)	—	—	—	—	—	—
	PA14 I/O (Port)	$\overline{WRHH}$ output, DQMUU output (BSC)	$\overline{RASL}$ output (BSC)	—	—	—	—	—
	PA13 I/O (Port)	$\overline{WRHL}$ output, DQMUL output (BSC)	$\overline{CASL}$ output (BSC)	—	—	—	—	—
	PA12 I/O (Port)	$\overline{CS0}$ output (BSC)	—	IRQ0 input (INTC)	TIC5U input (MTU2)	SSL1 output (RSPI)	—	TX_CLK input (Ether)
	PA11 I/O (Port)	$\overline{CS1}$ output (BSC)	—	IRQ1 input (INTC)	TIC5V input (MTU2)	CRx0 input (RCAN-ET)	RXD0 input (SCI)	TX_EN output (Ether)
	PA10 I/O (Port)	$\overline{CS2}$ output (BSC)	—	IRQ2 input (INTC)	TIC5W input (MTU2)	CTx0 output (RCAN-ET)	TXD0 output (SCI)	MII_TXD0 output (Ether)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
A	PA9 I/O (Port)	$\overline{CS3}$ output (BSC)	—	IRQ3 input (INTC)	TCLKD input (MTU2)	SSLO I/O (RSPI)	SCK0 I/O (SCI)	MII_TXD1 output (Ether)
	PA8 I/O (Port)	$\overline{CS4}$ output (BSC)	—	IRQ4 input (INTC)	TCLKC input (MTU2)	MISO I/O (RSPI)	RXD1 input (SCI)	MII_TXD2 output (Ether)
	PA7 I/O (Port)	$\overline{CS5}$ output (BSC)	—	IRQ5 input (INTC)	TCLKB input (MTU2)	MOSI I/O (RSPI)	TXD1 output (SCI)	MII_TXD3 output (Ether)
	PA6 I/O (Port)	$\overline{CS6}$ output (BSC)	—	IRQ6 input (INTC)	TCLKA input (MTU2)	RSPCK I/O (RSPI)	SCK1 I/O (SCI)	TX_ER output (Ether)
	PA5 I/O (Port)	$\overline{CS5}$ output (BSC)	—	—	TCLKA input (MTU2)	RSPCK I/O (RSPI)	SCK1 I/O (SCI)	RX_ER input (Ether)
	PA4 I/O (Port)	$\overline{CS4}$ output (BSC)	—	—	TCLKB input (MTU2)	MOSI I/O (RSPI)	TXD1 output (SCI)	MII_RXD3 input (Ether)
	PA3 I/O (Port)	$\overline{CS3}$ output (BSC)	—	—	TCLKC input (MTU2)	MISO I/O (RSPI)	RXD1 input (SCI)	MII_RXD2 input (Ether)
	PA2 I/O (Port)	$\overline{CS2}$ output (BSC)	—	—	TCLKD input (MTU2)	SSLO I/O (RSPI)	SCK0 I/O (SCI)	MII_RXD1 input (Ether)
	PA1 I/O (Port)	$\overline{CS1}$ output (BSC)	—	IRQ5 input (INTC)	—	CTx0 output (RCAN-ET)	TXD0 output (SCI)	MII_RXD0 input (Ether)
	PA0 I/O (Port)	$\overline{CS0}$ output (BSC)	—	IRQ4 input (INTC)	—	CRx0 input (RCAN-ET)	RXD0 input (SCI)	RX_CLK input (Ether)

Table 22.2 Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
B	PB15 I/O (Port)	—	—	IRQ7 input (INTC)	—	—	—	—
	PB14 I/O (Port)	—	—	IRQ6 input (INTC)	—	—	—	—
	PB13 input (Port)	—	—	IRQ3 input (INTC)	—	$\overline{\text{POE2}}$ input (POE2)	SDA I/O (IIC3)	—
	PB12 input (Port)	—	—	IRQ2 input (INTC)	—	$\overline{\text{POE1}}$ input (POE2)	SCL I/O (IIC3)	—
	PB11 I/O (Port)	$\overline{\text{CS1}}$ output (BSC)	$\overline{\text{CS3}}$ output (BSC)	IRQ1 input (INTC)	—	—	TXD2 output (SCI)	$\overline{\text{CS7}}$ output (BSC)
	PB10 I/O (Port)	$\overline{\text{CS0}}$ output (BSC)	$\overline{\text{CS2}}$ output (BSC)	IRQ0 input (INTC)	—	—	RXD2 input (SCI)	$\overline{\text{CS6}}$ output (BSC)
	PB9 I/O (Port)	A25 output (BSC)	DACK0 output (DMAC)	—	TCLKA input (MTU2)	—	TXD4 output (SCI)	$\overline{\text{CS3}}$ output (BSC)
	PB8 I/O (Port)	A24 output (BSC)	DREQ0 input (DMAC)	—	TCLKB input (MTU2)	—	RXD4 input (SCI)	$\overline{\text{CS2}}$ output (BSC)
	PB7 I/O (Port)	A23 output (BSC)	TEND0 output (DMAC)	IRQ7 input (INTC)	TCLKC input (MTU2)	—	SCK4 I/O (SCI)	RD/$\overline{\text{WR}}$ output (BSC)
	PB6 I/O (Port)	A22 output (BSC)	$\overline{\text{WAIT}}$ input (BSC)	IRQ6 input (INTC)	TCLKD input (MTU2)	—	TXD0 output (SCI)	—
	PB5 I/O (Port)	A21 output (BSC)	$\overline{\text{BREQ}}$ input (BSC)	IRQ5 input (INTC)	—	—	RXD0 input (SCI)	—
	PB4 I/O (Port)	A20 output (BSC)	$\overline{\text{BACK}}$ output (BSC)	IRQ4 input (INTC)	TIOC0D I/O (MTU2)	$\overline{\text{WAIT}}$ input (BSC)	SCK3 I/O (SCIF)	$\overline{\text{BS}}$ output (BSC)
	PB3 I/O (Port)	A19 output (BSC)	$\overline{\text{BREQ}}$ input (BSC)	IRQ3 input (INTC)	TIOC0C I/O (MTU2)	$\overline{\text{CASL}}$ output (BSC)	TXD3 output (SCIF)	$\overline{\text{AH}}$ output (BSC)
	PB2 I/O (Port)	A18 output (BSC)	$\overline{\text{BACK}}$ output (BSC)	IRQ2 input (INTC)	TIOC0B I/O (MTU2)	$\overline{\text{RASL}}$ output (BSC)	RXD3 input (SCIF)	$\overline{\text{FRAME}}$ output (BSC)
	PB1 I/O (Port)	A17 output (BSC)	$\overline{\text{IRQOUT}}$ output (INTC)/ $\overline{\text{REFOUT}}$ output (BSC)	IRQ1 input (INTC)	TIOC0A I/O (MTU2)	—	—	$\overline{\text{ADTRG}}$ input (ADC)
	PB0 I/O (Port)	A16 output (BSC)	RD/$\overline{\text{WR}}$ output (BSC)	IRQ0 input (INTC)	TIOC2A I/O (MTU2)	—	—	—

Table 22.3 Multiplexed Pins (Port C)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
C	PC15 I/O (Port)	A15 output (BSC)	—	IRQ2 input (INTC)	TCLKD input (MTU2)	—	—	—
	PC14 I/O (Port)	A14 output (BSC)	—	IRQ1 input (INTC)	TCLKC input (MTU2)	—	—	—
	PC13 I/O (Port)	A13 output (BSC)	—	IRQ0 input (INTC)	TCLKB input (MTU2)	—	—	—
	PC12 I/O (Port)	A12 output (BSC)	—	—	TCLKA input (MTU2)	—	—	—
	PC11 I/O (Port)	A11 output (BSC)	—	—	TIOC1B I/O (MTU2)	CTx0 output (RCAN-ET)	TXD0 output (SCI)	—
	PC10 I/O (Port)	A10 output (BSC)	—	—	TIOC1A I/O (MTU2)	CRx0 input (RCAN-ET)	RXD0 input (SCI)	—
	PC9 I/O (Port)	A9 output (BSC)	—	—	—	CTx0 output (RCAN-ET)	TXD0 output (SCI)	—
	PC8 I/O (Port)	A8 output (BSC)	—	—	—	CRx0 input (RCAN-ET)	RXD0 input (SCI)	—
	PC7 I/O (Port)	A7 output (BSC)	—	—	—	—	—	—
	PC6 I/O (Port)	A6 output (BSC)	—	—	—	—	—	—
	PC5 I/O (Port)	A5 output (BSC)	—	—	—	—	—	—
	PC4 I/O (Port)	A4 output (BSC)	—	—	—	—	—	—
	PC3 I/O (Port)	A3 output (BSC)	—	—	—	—	—	—
	PC2 I/O (Port)	A2 output (BSC)	—	—	—	—	—	—
	PC1 I/O (Port)	A1 output (BSC)	—	—	—	—	—	—
	PC0 I/O (Port)	A0 output (BSC)	—	IRQ4 input (INTC)	—	$\overline{\text{POE0}}$ input (POE2)	—	—

Table 22.4 Multiplexed Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
D	PD31 I/O (Port)	D31 I/O (BSC)	—	—	—	TIOC3AS I/O (MTU2S)	SSL2 output (RSPI)	RX_DV input (Ether)
	PD30 I/O (Port)	D30 I/O (BSC)	—	—	—	TIOC3CS I/O (MTU2S)	SSL3 output (RSPI)	RX_ER input (Ether)
	PD29 I/O (Port)	D29 I/O (BSC)	—	—	—	TIOC3BS I/O (MTU2S)	—	MII_RXD3 input (Ether)
	PD28 I/O (Port)	D28 I/O (BSC)	—	—	—	TIOC3DS I/O (MTU2S)	—	MII_RXD2 input (Ether)
	PD27 I/O (Port)	D27 I/O (BSC)	—	—	—	TIOC4AS I/O (MTU2S)	—	MII_RXD1 input (Ether)
	PD26 I/O (Port)	D26 I/O (BSC)	—	—	—	TIOC4BS I/O (MTU2S)	—	MII_RXD0 input (Ether)
	PD25 I/O (Port)	D25 I/O (BSC)	—	—	—	TIOC4CS I/O (MTU2S)	—	RX_CLK input (Ether)
	PD24 I/O (Port)	D24 I/O (BSC)	—	—	—	TIOC4DS I/O (MTU2S)	—	CRS input (Ether)
	PD23 I/O (Port)	D23 I/O (BSC)	DACK1 output (DMAC)	IRQ7 input (INTC)	—	—	—	COL input (Ether)
	PD22 I/O (Port)	D22 I/O (BSC)	DREQ1 input (DMAC)	IRQ6 input (INTC)	—	—	—	WOL output (Ether)
	PD21 I/O (Port)	D21 I/O (BSC)	TEND1 output (DMAC)	IRQ5 input (INTC)	AUDCK output (AUD)	—	—	EXOUT output (Ether)
	PD20 I/O (Port)	D20 I/O (BSC)	—	IRQ4 input (INTC)	$\overline{\text{AUDSYNC}}$ output (AUD)	—	—	MDC output (Ether)
	PD19 I/O (Port)	D19 I/O (BSC)	—	IRQ3 input (INTC)	AUDATA3 output (AUD)	—	—	LNKSTA input (Ether)
	PD18 I/O (Port)	D18 I/O (BSC)	—	IRQ2 input (INTC)	AUDATA2 output (AUD)	—	—	MDIO I/O (Ether)
	PD17 I/O (Port)	D17 I/O (BSC)	—	IRQ1 input (INTC)	AUDATA1 output (AUD)	$\overline{\text{POE4}}$ input (POE2)	—	$\overline{\text{ADTRG}}$ input (ADC)
	PD16 I/O (Port)	D16 I/O (BSC)	$\overline{\text{UBCTRG}}$ output (UBC)	IRQ0 input (INTC)	AUDATA0 output (AUD)	$\overline{\text{POE0}}$ input (POE2)	—	—
	PD15 I/O (Port)	D15 I/O (BSC)	—	—	—	TIOC4DS I/O (MTU2S)	—	—

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
D	PD14 I/O (Port)	D14 I/O (BSC)	—	—	—	TIOC4CS I/O (MTU2S)	—	—
	PD13 I/O (Port)	D13 I/O (BSC)	—	—	—	TIOC4BS I/O (MTU2S)	—	—
	PD12 I/O (Port)	D12 I/O (BSC)	—	—	—	TIOC4AS I/O (MTU2S)	—	—
	PD11 I/O (Port)	D11 I/O (BSC)	—	—	—	TIOC3DS I/O (MTU2S)	—	—
	PD10 I/O (Port)	D10 I/O (BSC)	—	—	—	TIOC3BS I/O (MTU2S)	—	—
	PD9 I/O (Port)	D9 I/O (BSC)	—	—	—	TIOC3CS I/O (MTU2S)	—	—
	PD8 I/O (Port)	D8 I/O (BSC)	—	—	—	TIOC3AS I/O (MTU2S)	—	—
	PD7 I/O (Port)	D7 I/O (BSC)	—	—	—	TIC5WS input (MTU2S)	—	—
	PD6 I/O (Port)	D6 I/O (BSC)	—	—	—	TIC5VS input (MTU2S)	—	—
	PD5 I/O (Port)	D5 I/O (BSC)	—	—	—	TIC5US input (MTU2S)	—	—
	PD4 I/O (Port)	D4 I/O (BSC)	—	—	TIC5W input (MTU2)	—	SCK2 I/O (SCI)	—
	PD3 I/O (Port)	D3 I/O (BSC)	—	—	TIC5V input (MTU2)	—	TXD2 output (SCI)	—
	PD2 I/O (Port)	D2 I/O (BSC)	—	—	TIC5U input (MTU2)	—	RXD2 input (SCI)	—
	PD1 I/O (Port)	D1 I/O (BSC)	—	—	—	—	—	—
	PD0 I/O (Port)	D0 I/O (BSC)	—	—	—	—	—	—

Table 22.5 Multiplexed Pins (Port E)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
E	PE15 I/O (Port)	—	DACK1 output (DMAC)	$\overline{\text{IRQOUT}}$ output (INTC)/ $\overline{\text{REFOUT}}$ output (BSC)	TIOC4D I/O (MTU2)	—	—	TX_ER output (Ether)
	PE14 I/O (Port)	—	DACK0 output (DMAC)	—	TIOC4C I/O (MTU2)	—	—	MII_TXD3 output (Ether)
	PE13 I/O (Port)	—	—	$\overline{\text{MRES}}$ input (system control)	TIOC4B I/O (MTU2)	—	—	MII_TXD2 output (Ether)
	PE12 I/O (Port)	—	—	—	TIOC4A I/O (MTU2)	—	—	MII_TXD1 output (Ether)
	PE11 I/O (Port)	—	DACK3 output (DMAC)	—	TIOC3D I/O (MTU2)	—	—	MII_TXD0 output (Ether)
	PE10 I/O (Port)	—	DREQ3 input (DMAC)	—	TIOC3C I/O (MTU2)	SSL3 output (RSPI)	TXD2 output (SCI)	TX_CLK input (Ether)
	PE9 I/O (Port)	—	DACK2 output (DMAC)	—	TIOC3B I/O (MTU2)	—	—	TX_EN output (Ether)
	PE8 I/O (Port)	—	DREQ2 input (DMAC)	—	TIOC3A I/O (MTU2)	SSL2 output (RSPI)	SCK2 I/O (SCI)	EXOUT output (Ether)
	PE7 I/O (Port)	—	$\overline{\text{UBCTRG}}$ output (UBC)	—	TIOC2B I/O (MTU2)	SSL1 output (RSPI)	RXD2 input (SCI)	RX_DV input (Ether)
	PE6 I/O (Port)	—	—	—	TIOC2A I/O (MTU2)	TIOC3DS I/O (MTU2S)	RXD3 input (SCIF)	—
	PE5 I/O (Port)	—	—	—	TIOC1B I/O (MTU2)	TIOC3BS I/O (MTU2S)	TXD3 output (SCIF)	MDIO I/O (Ether)
	PE4 I/O (Port)	—	—	IRQ4 input (INTC)	TIOC1A I/O (MTU2)	$\overline{\text{POE8}}$ input (POE2)	SCK3 I/O (SCIF)	CRS input (Ether)
	PE3 I/O (Port)	—	TEND1 output (DMAC)	—	TIOC0D I/O (MTU2)	TIOC4DS I/O (MTU2S)	—	COL input (Ether)
	PE2 I/O (Port)	—	DREQ1 input (DMAC)	—	TIOC0C I/O (MTU2)	TIOC4CS I/O (MTU2S)	—	WOL output (Ether)
	PE1 I/O (Port)	—	TEND0 output (DMAC)	—	TIOC0B I/O (MTU2)	TIOC4BS I/O (MTU2S)	—	MDC output (Ether)
	PE0 I/O (Port)	—	DREQ0 input (DMAC)	—	TIOC0A I/O (MTU2)	TIOC4AS I/O (MTU2S)	—	LNKSTA input (Ether)

Table 22.6 Multiplexed Pins (Port F)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)	Function 8 (Related Module)
F	PF7 input (Port)	AN7 input (ADC)	—	—	—	—	—	—
	PF6 input (Port)	AN6 input (ADC)	—	—	—	—	—	—
	PF5 input (Port)	AN5 input (ADC)	—	—	—	—	—	—
	PF4 input (Port)	AN4 input (ADC)	—	—	—	—	—	—
	PF3 input (Port)	AN3 input (ADC)	—	—	—	—	—	—
	PF2 input (Port)	AN2 input (ADC)	—	—	—	—	—	—
	PF1 input (Port)	AN1 input (ADC)	—	—	—	—	—	—
	PF0 input (Port)	AN0 input (ADC)	—	—	—	—	—	—

Note: AN input function is valid during A/D conversion.

Table 22.7 List of pin functions in each operating mode

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
B3, G3, M3, P4, D5, N7, N12, J13, C14, M14	19, 38, 51, 65, 85, 95, 104, 130, 163, 174	VccQ				—
A2, G2, M2, R3, E4, F4, K4, A6, M6, A8, M9, P10, L12, R14, B15, F15, H15, K15	8, 13, 20, 29, 39, 50, 56, 66, 76, 86, 96, 105, 108, 120, 131, 156, 164, 175	Vss				—
E12	124	PLLVcc				—
E13	122	PLLVss				—
H14	112	DrVcc (VCCQ)				—
G12	115	DrVss				—
N1, D3, P3, C8, N10, J12, B14	7, 40, 49, 75, 106, 132, 155	V_{CL}				—
C10, C11	142, 145	AVcc				—
B9, D12	137, 150	AVss				—
A11, B11	143, 144	AVref				—
A9, A14	136, 151	AVrefVss				—
E14	121	EXTAL				—
E15	119	XTAL				—
J14	109	USBEXTAL				—
J15	107	USBXTAL				—
C9	152	MD0				—
D8	153	MD1				—
A15	133	$\overline{RES}$				—
D7	154	$\overline{WDTOVF}$				—

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
F13	123	NMI				—
C13	134	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$				—
B13	135	$\overline{\text{ASEMD0}}$				—
D13	127	TCK				—
D14	128	TMS				—
D15	125	TDI				—
C15	126	TDO				—
C12	129	$\overline{\text{TRST}}$				—
B8	157	PA0				PA0/$\overline{\text{CS0}}$*1/IRQ4/CRx0/RXD0/RX_CLK
C7	158	PA1				PA1/$\overline{\text{CS1}}$*1/IRQ5/CTx0/TXD0/MII_RXD0
A7	159	PA2				PA2/$\overline{\text{CS2}}$*1/TCLKD/SSL0/SCK0/MII_RXD1
B7	160	PA3				PA3/$\overline{\text{CS3}}$*1/TCLKC/MISO/RXD1/MII_RXD2
D6	161	PA4				PA4/$\overline{\text{CS4}}$*1/TCLKB/MOSI/TXD1/MII_RXD3
C6	162	PA5				PA5/$\overline{\text{CS5}}$*1/TCLKA/RSPCK/SCK1/RX_ER
K14	103	PA6				PA6/$\overline{\text{CS6}}$*1/IRQ6/TCLKA/RSPCK/SCK1/TX_ER
K13	102	PA7				PA7/$\overline{\text{CS5}}$*1/IRQ5/TCLKB/MOSI/TXD1/MII_TXD3
K12	101	PA8				PA8/$\overline{\text{CS4}}$*1/IRQ4/TCLKC/MISO/RXD1/MII_TXD2
L15	100	PA9				PA9/$\overline{\text{CS3}}$*1/IRQ3/TCLKD/SSL0/SCK0/MII_TXD1

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
L14	99			PA10		PA10/$\overline{CS2}$*1/IRQ2/TIC5W/ CTx0/TXD0/MII_TXD0
L13	98			PA11		PA11/$\overline{CS1}$*1/IRQ1/TIC5V/ CRx0/RXD0/TX_EN
M15	97			PA12		PA12/$\overline{CS0}$*1/IRQ0/TIC5U/ SSL1/TX_CLK
G1	18	$\overline{WRHL}$/DQMUL		PA13		PA13/$\overline{WRHL}$*1/DQMUL*1/ $\overline{CASL}$*1
G4	17	$\overline{WRHH}$/DQMUU		PA14		PA14/$\overline{WRHH}$*1/DQMUU*1/ $\overline{RASL}$*1
F2	16	$\overline{WRH}$/DQMLU	$\overline{WRH}$/DQMLU	PA15	PA15	PA15/$\overline{WRH}$*1/DQMLU*1
F1	15	$\overline{WRL}$/DQMLL	$\overline{WRL}$/DQMLL	PA16	PA16	PA16/$\overline{WRL}$*1/DQMLL*1
F3	14	$\overline{RD}$	$\overline{RD}$	PA17	PA17	PA17/$\overline{RD}$*1
E1	12	CK	CK	CK	PA18	PA18/CK
E2	11			PA19		PA19/$\overline{WRH}$*1/DQMLU*1/ $\overline{WAIT}$*1/IRQ7/$\overline{RASU}$*1/$\overline{POE8}$/ RXD1/$\overline{BS}$*1
E3	10			PA20		PA20/$\overline{WRL}$*1/DQMLL*1/ $\overline{BREQ}$*1/IRQ6/$\overline{CASU}$*1/$\overline{POE4}$/ TXD1/$\overline{AH}$*1
D1	9			PA21		PA21/$\overline{RD}$*1/$\overline{BACK}$*1/IRQ5/ CKE*1/$\overline{POE3}$/SCK1/$\overline{FRAME}$*1
M4	41	A16	A16	PB0	PB0	PB0/A16*1/RD/$\overline{WR}$*1/IRQ0/ TIOC2A
N2	42	A17	A17	PB1	PB1	PB1/A17*1/$\overline{IRQOUT}$/ $\overline{REFOUT}$*1/IRQ1/TIOC0A/ $\overline{ADTRG}$

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
P1	43	A18	A18	PB2	PB2	PB2/A18*1/$\overline{\text{BACK}}$*1/IRQ2/ TIOC0B/$\overline{\text{RASL}}$*1/RXD3/ $\overline{\text{FRAME}}$*1
P2	44	A19	A19	PB3	PB3	PB3/A19*1/$\overline{\text{BREQ}}$*1/IRQ3/ TIOC0C/$\overline{\text{CASL}}$*1/TXD3/$\overline{\text{AH}}$*1
R1	45	A20	A20	PB4	PB4	PB4/A20*1/$\overline{\text{BACK}}$*1/IRQ4/ TIOC0D/$\overline{\text{WAIT}}$*1/SCK3/$\overline{\text{BS}}$*1
N3	46	A21	A21	PB5	PB5	PB5/A21*1/$\overline{\text{BREQ}}$*1/IRQ5/ RXD0
R2	47	A22	A22	PB6	PB6	PB6/A22*1/$\overline{\text{WAIT}}$*1/IRQ6/ TCLKD/TXD0
N4	48	A23	A23	PB7	PB7	PB7/A23*1/TEND0/IRQ7/ TCLKC/SCK4/RD/$\overline{\text{WR}}$*1
M5	52	A24	A24	PB8	PB8	PB8/A24*1/DREQ0/TCLKB/ RXD4/$\overline{\text{CS2}}$*1
R4	53	A25	A25	PB9	PB9	PB9/A25*1/DACK0/TCLKA/ TXD4/$\overline{\text{CS3}}$*1
N5	54	$\overline{\text{CS0}}$	$\overline{\text{CS0}}$	PB10	PB10	PB10/$\overline{\text{CS0}}$*1/$\overline{\text{CS2}}$*1/IRQ0/ RXD2/$\overline{\text{CS6}}$*1
P5	55	$\overline{\text{CS1}}$	$\overline{\text{CS1}}$	PB11	PB11	PB11/$\overline{\text{CS1}}$*1/$\overline{\text{CS3}}$*1/IRQ1/ TXD2/$\overline{\text{CS7}}$*1
H12	110	PB12				PB12/IRQ2/$\overline{\text{POE1}}$/SCL
H13	111	PB13				PB13/IRQ3/$\overline{\text{POE2}}$/SDA
F12	116	PB14				PB14/IRQ6
F14	117	PB15				PB15/IRQ7
H4	21	A0	A0	PC0	PC0	PC0/A0*1/IRQ4/$\overline{\text{POE0}}$
H3	22	A1	A1	PC1	PC1	PC1/A1*1
H1	23	A2	A2	PC2	PC2	PC2/A2*1
H2	24	A3	A3	PC3	PC3	PC3/A3*1

Pin name						
		Initial function				
Pin number	Pin number	On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
J4	25	A4	A4	PC4	PC4	PC4/A4*1
J3	26	A5	A5	PC5	PC5	PC5/A5*1
J1	27	A6	A6	PC6	PC6	PC6/A6*1
J2	28	A7	A7	PC7	PC7	PC7/A7*1
K3	30	A8	A8	PC8	PC8	PC8/A8*1/CRx0/RXD0
K1	31	A9	A9	PC9	PC9	PC9/A9*1/CTx0/TXD0
K2	32	A10	A10	PC10	PC10	PC10/A10*1/TIOC1A/CRx0/ RXD0
L3	33	A11	A11	PC11	PC11	PC11/A11*1/TIOC1B/CTx0/ TXD0
L1	34	A12	A12	PC12	PC12	PC12/A12*1/TCLKA
L2	35	A13	A13	PC13	PC13	PC13/A13*1/IRQ0/TCLKB
L4	36	A14	A14	PC14	PC14	PC14/A14*1/IRQ1/TCLKC
M1	37	A15	A15	PC15	PC15	PC15/A15*1/IRQ2/TCLKD
R5	57	D0	D0	PD0	PD0	PD0/D0*1
N6	58	D1	D1	PD1	PD1	PD1/D1*1
R6	59	D2	D2	PD2	PD2	PD2/D2*1/TIC5U/RXD2
P6	60	D3	D3	PD3	PD3	PD3/D3*1/TIC5V/TXD2
M7	61	D4	D4	PD4	PD4	PD4/D4*1/TIC5W/SCK2
M8	62	D5	D5	PD5	PD5	PD5/D5*1/TIC5US
R7	63	D6	D6	PD6	PD6	PD6/D6*1/TIC5VS
P7	64	D7	D7	PD7	PD7	PD7/D7*1/TIC5WS
R8	67	D8	D8	PD8	PD8	PD8/D8*1/TIOC3AS
P8	68	D9	D9	PD9	PD9	PD9/D9*1/TIOC3CS
N8	69	D10	D10	PD10	PD10	PD10/D10*1/TIOC3BS
N9	70	D11	D11	PD11	PD11	PD11/D11*1/TIOC3DS
R9	71	D12	D12	PD12	PD12	PD12/D12*1/TIOC4AS
P9	72	D13	D13	PD13	PD13	PD13/D13*1/TIOC4BS
M10	73	D14	D14	PD14	PD14	PD14/D14*1/TIOC4CS

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
R10	74	D15	D15	PD15	PD15	PD15/D15*1/TIOC4DS
N11	77	D16	PD16	PD16	PD16	PD16/D16*1/$\overline{\text{UBCTRG}}$/ IRQ0/AUDATA0/$\overline{\text{POE0}}$
R11	78	D17	PD17	PD17	PD17	PD17/D17*1/IRQ1/ AUDATA1/$\overline{\text{POE4}}$/$\overline{\text{ADTRG}}$
P11	79	D18	PD18	PD18	PD18	PD18/D18*1/IRQ2/ AUDATA2/MDIO
M11	80	D19	PD19	PD19	PD19	PD19/D19*1/IRQ3/ AUDATA3/LNKSTA
R12	81	D20	PD20	PD20	PD20	PD20/D20*1/IRQ4/ $\overline{\text{AUDSYNC}}$/MDC
M12	82	D21	PD21	PD21	PD21	PD21/D21*1/TEND1/IRQ5/ AUDCK/EXOUT
P12	83	D22	PD22	PD22	PD22	PD22/D22*1/DREQ1/IRQ6/ WOL
R13	84	D23	PD23	PD23	PD23	PD23/D23*1/DACK1/IRQ7/ COL
P13	87	D24	PD24	PD24	PD24	PD24/D24*1/TIOC4DS/CRS
P14	88	D25	PD25	PD25	PD25	PD25/D25*1/TIOC4CS/ RX_CLK
R15	89	D26	PD26	PD26	PD26	PD26/D26*1/TIOC4BS/ MII_RXD0
N13	90	D27	PD27	PD27	PD27	PD27/D27*1/TIOC4AS/ MII_RXD1
P15	91	D28	PD28	PD28	PD28	PD28/D28*1/TIOC3DS/ MII_RXD2
N14	92	D29	PD29	PD29	PD29	PD29/D29*1/TIOC3BS/ MII_RXD3

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
M13	93	D30	PD30	PD30	PD30	PD30/D30*1/TIOC3CS/SSL3/RX_ER
N15	94	D31	PD31	PD31	PD31	PD31/D31*1/TIOC3AS/SSL2/RX_DV
B2	176	PE0				PE0/DREQ0/TIOC0A/TIOC4AS/LNKSTA
A1	1	PE1				PE1/TEND0/TIOC0B/TIOC4BS/MDC
C3	2	PE2				PE2/DREQ1/TIOC0C/TIOC4CS/WOL
B1	3	PE3				PE3/TEND1/TIOC0D/TIOC4DS/COL
C2	4	PE4				PE4/IRQ4/TIOC1A/$\overline{\text{POE8}}$/SCK3/CRS
D2	5	PE5				PE5/TIOC1B/TIOC3BS/TXD3/MDIO
C1	6	PE6				PE6/TIOC2A/TIOC3DS/RXD3
B6	165	PE7				PE7/$\overline{\text{UBCTRG}}$/TIOC2B/SSL1/RXD2/RX_DV
A5	166	PE8				PE8/DREQ2/TIOC3A/SSL2/SCK2/EXOUT
B5	167	PE9				PE9/DACK2/TIOC3B/TX_EN
C5	168	PE10				PE10/DREQ3/TIOC3C/SSL3/TXD2/TX_CLK
A4	169	PE11				PE11/DACK3/TIOC3D/MII_TXD0
C4	170	PE12				PE12/TIOC4A/MII_TXD1
B4	171	PE13				PE13/$\overline{\text{MRES}}$/TIOC4B/MII_TXD2

Pin number	Pin number	Pin name				
		Initial function				
		On-chip ROM unabled mode		On-chip ROM enabled mode	Single-chip mode	
BGA	LQFP	MCU mode 0	MCU mode 1	MCU mode 2	MCU mode 3	Settable function in PFC
A3	172	PE14				PE14/DACK0/TIOC4C/ MII_TXD3
D4	173	PE15				PE15/DACK1/$\overline{\text{IRQOUT}}$/ $\overline{\text{REFOUT}}$*1/TIOC4D/TX_ER
A13	138	PF0/AN0				—*2
B12	139	PF1/AN1				—*2
D11	140	PF2/AN2				—*2
A12	141	PF3/AN3				—*2
D10	146	PF4/AN4				—*2
A10	147	PF5/AN5				—*2
B10	148	PF6/AN6				—*2
D9	149	PF7/AN7				—*2
G15	113	USD+				—
G14	114	USD-				—
G13	118	VBUS				—

Notes: 1. This function is enabled in only on-chip ROM enabled/disabled external extension mode. Do not set it in single-chip mode.

2. A pin function is analog input during sampling by the A/D converter, and general input during a period other than this sampling.

22.1 Register Descriptions

The PFC has the following registers. See section 32, List of Registers for register addresses and register states in each operating mode.

Table 22.8 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A I/O register H	PAIORH	R/W	H'0000	H'FFFE3804	8, 16, 32
Port A I/O register L	PAIORL	R/W	H'0000	H'FFFE3806	8, 16
Port A control register H2	PACRH2	R/W	H'0000	H'FFFE380C	8, 16, 32
Port A control register H1	PACRH1	R/W	H'0000*	H'FFFE380E	8, 16
Port A control register L4	PACRL4	R/W	H'0000*	H'FFFE3810	8, 16, 32
Port A control register L3	PACRL3	R/W	H'0000	H'FFFE3812	8, 16
Port A control register L2	PACRL2	R/W	H'0000	H'FFFE3814	8, 16, 32
Port A control register L1	PACRL1	R/W	H'0000	H'FFFE3816	8, 16
Port A pull-up MOS control register H	PAPCRH	R/W	H'0000	H'FFFE3828	8, 16, 32
Port A pull-up MOS control register L	PAPCRL	R/W	H'0000	H'FFFE382A	8, 16
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE3886	8, 16
Port B control register L4	PBCRL4	R/W	H'0000	H'FFFE3890	8, 16, 32
Port B control register L3	PBCRL3	R/W	H'0000*	H'FFFE3892	8, 16
Port B control register L2	PBCRL2	R/W	H'0000*	H'FFFE3894	8, 16, 32
Port B control register L1	PBCRL1	R/W	H'0000*	H'FFFE3896	8, 16
Port B pull-up MOS control register L	PBPCRL	R/W	H'0000	H'FFFE38AA	8, 16
Port C I/O register L	PCIORL	R/W	H'0000	H'FFFE3906	8, 16
Port C control register L4	PCCRL4	R/W	H'0000*	H'FFFE3910	8, 16, 32
Port C control register L3	PCCRL3	R/W	H'0000*	H'FFFE3912	8, 16
Port C control register L2	PCCRL2	R/W	H'0000*	H'FFFE3914	8, 16, 32
Port C control register L1	PCCRL1	R/W	H'0000*	H'FFFE3916	8, 16
Port C pull-up MOS control register L	PCPCRL	R/W	H'0000	H'FFFE392A	8, 16
Port D I/O register H	PDIORH	R/W	H'0000	H'FFFE3984	8, 16, 32
Port D I/O register L	PDIORL	R/W	H'0000	H'FFFE3986	8, 16
Port D control register H4	PDCRH4	R/W	H'0000*	H'FFFE3988	8, 16, 32
Port D control register H3	PDCRH3	R/W	H'0000*	H'FFFE398A	8, 16
Port D control register H2	PDCRH2	R/W	H'0000*	H'FFFE398C	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D control register H1	PDCRH1	R/W	H'0000*	H'FFFE398E	8, 16
Port D control register L4	PDCRL4	R/W	H'0000*	H'FFFE3990	8, 16, 32
Port D control register L3	PDCRL3	R/W	H'0000*	H'FFFE3992	8, 16
Port D control register L2	PDCRL2	R/W	H'0000*	H'FFFE3994	8, 16, 32
Port D control register L1	PDCRL1	R/W	H'0000*	H'FFFE3996	8, 16
Port D pull-up MOS control register H	PDPCRH	R/W	H'0000	H'FFFE39A8	8, 16, 32
Port D pull-up MOS control register L	PDPCRL	R/W	H'0000	H'FFFE39AA	8, 16
Port E I/O register L	PEIORL	R/W	H'0000	H'FFFE3A06	8, 16
Port E control register L4	PECRL4	R/W	H'0000	H'FFFE3A10	8, 16, 32
Port E control register L3	PECRL3	R/W	H'0000	H'FFFE3A12	8, 16
Port E control register L2	PECRL2	R/W	H'0000	H'FFFE3A14	8, 16, 32
Port E control register L1	PECRL1	R/W	H'0000	H'FFFE3A16	8, 16
Large current port control register	HCPCR	R/W	H'000F	H'FFFE3A20	8, 16, 32
IRQOUT function control register	IFCR	R/W	H'0000	H'FFFE3A22	8, 16
Port E pull-up MOS control register L	PEPCRL	R/W	H'0000	H'FFFE3A2A	8, 16
DACK output timing control register	PDACKCR	R/W	H'0000	H'FFFE3A2C	8, 16

Note: * The initial values of registers in each product vary according to the setting of the operating mode. See the description of each register in this section for details.

22.1.1 Port A I/O Registers H and L (PAIORH and PAIORL)

PAIORH and PAIORL are 16-bit readable/writable registers that are used to set the pins on port A as inputs or outputs. Bits PA21IOR to PA01IOR correspond to pins PA21 to PA0 (multiplexed port pin names except for the port names are abbreviated here). PAIORH and PAIORL are enabled when the port A pins are functioning as general-purpose inputs/outputs (PA21 to PA16 for PAIORH and PA15 to PA0 for PAIORL). In other states, they are disabled. A given pin on port A will be an output pin if the corresponding bit in PAIORH or PAIORL is set to 1, and an input pin if the bit is cleared to 0. Bits 15 to 6 of PAIORH are reserved. These bits are always read as 0. The write value should always be 0.

The initial values of PAIORL and PAIORH are both H'0000.

- Port A I/O Register H (PAIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA21 IOR	PA20 IOR	PA19 IOR	PA18 IOR	PA17 IOR	PA16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Port A I/O Register L (PAIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.2 Port A Control Registers H1 and H2, and L1 to L4 (PACRH1 and PACRH2, and PACRL1 to PACRL4)

PACRH1 and PACRH2, and PACRL1 to PACRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port A.

- Port A Control Register H2 (PACRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PA21MD[2:0]			-	PA20MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 4	PA21MD[2:0]	000	R/W	PA21 Mode Select the function of the PA21/$\overline{\text{RD}}$/$\overline{\text{BACK}}$/IRQ5/CKE/$\overline{\text{POE3}}$/SCK1/$\overline{\text{FRAME}}$ pin. 000: PA21 I/O (port) 001: $\overline{\text{RD}}$ output (BSC) 010: $\overline{\text{BACK}}$ output (BSC) 011: IRQ5 input (INTC) 100: CKE output (BSC) 101: $\overline{\text{POE3}}$ input (POE2) 110: SCK1 I/O (SCI) 111: $\overline{\text{FRAME}}$ output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PA20MD[2:0]	000	R/W	PA20 Mode Select the function of the PA20/$\overline{\text{WRL}}$/DQMLL/$\overline{\text{BREQ}}$/IRQ6/$\overline{\text{CASU}}$/$\overline{\text{POE4}}$/TXD1/$\overline{\text{AH}}$ pin. 000: PA20 I/O (port) 001: $\overline{\text{WRL}}$ output, DQMLL output (BSC) 010: $\overline{\text{BREQ}}$ input (BSC) 011: IRQ6 input (INTC) 100: $\overline{\text{CASU}}$ output (BSC) 101: $\overline{\text{POE4}}$ input (POE2) 110: TXD1 output (SCI) 111: $\overline{\text{AH}}$ output (BSC)

- Port A Control Register H1 (PACRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA19MD[2:0]			-	PA18MD[2:0]			-	PA17MD[2:0]			-	PA16MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0*2	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Notes: 1. The initial value is 1 during the on-chip ROM disabled external extension mode.
2. The initial value is 1 during the on-chip ROM enabled/disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PA19MD[2:0]	000	R/W	PA19 Mode Select the function of the PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$ output pin. 000: PA19 I/O (port) 001: $\overline{\text{WRH}}$ output, DQMLU output (BSC) 010: $\overline{\text{WAIT}}$ input (BSC) 011: IRQ7 input (INTC) 100: $\overline{\text{RASU}}$ output (BSC) 101: $\overline{\text{POE8}}$ input (POE2) 110: RXD1 input (SCI) 111: $\overline{\text{BS}}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA18MD[2:0]	000*2	R/W	PA18 Mode Select the function of the PA18/CK pin. 000: PA18 I/O (port) 001: CK output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PA17MD[2:0]	000*[1]	R/W	PA17 Mode Select the function of the PA17/$\overline{RD}$ pin. 000: PA17 I/O (port) 001: $\overline{RD}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PA16MD[2:0]	000*[1]	R/W	PA16 Mode Select the function of the PA16/$\overline{WRL}$/DQMLL pin. 000: PA16 I/O (port) 001: $\overline{WRL}$ output, DQMLL output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Notes: 1. The initial value is 001 during the on-chip ROM disabled external extension mode.
2. The initial value is 001 during the on-chip ROM enabled/disabled external extension mode.

- Port A Control Register L4 (PACRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15MD[2:0]			-	PA14MD[2:0]			-	PA13MD[2:0]			-	PA12MD[2:0]		
Initial value:	0	0	0	0*1	0	0	0	0*2	0	0	0	0*2	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Notes: 1. The initial value is 1 during the on-chip ROM disabled external extension mode.
2. The initial value is 1 during the on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA15MD[2:0]	000*1	R/W	PA15 Mode Select the function of the PA15/$\overline{\text{WRH}}$/DQMLU pin. 000: PA15 I/O (port) 001: $\overline{\text{WRH}}$ output, DQMLU output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA14MD[2:0]	000*2	R/W	PA14 Mode Select the function of the PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$ pin. 000: PA14 I/O (port) 001: $\overline{\text{WRHH}}$ output, DQMUU output (BSC) 010: $\overline{\text{RASL}}$ output (BSC) 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PA13MD[2:0]	000*2	R/W	PA13 Mode Select the function of the PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$ pin. 000: PA13 I/O (port) 001: $\overline{\text{WRHL}}$ output, DQMUL output (BSC) 010: $\overline{\text{CASL}}$ output (BSC) 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PA12MD[2:0]	000	R/W	PA12 Mode Select the function of the PA12/$\overline{\text{CS0}}$/IRQ0/TIC5U/SSL1/TX_CLK pin. 000: PA12 I/O (port) 001: $\overline{\text{CS0}}$ output (BSC) 010: Setting prohibited 011: IRQ0 input (INTC) 100: TIC5U input (MTU2) 101: SSL1 output (RSPI) 110: Setting prohibited 111: TX_CLK input (Ether)

Notes: 1. The initial value is 001 during the on-chip ROM disabled external extension mode.
2. The initial value is 001 during the on-chip ROM disabled 32-bit external extension mode.

- Port A Control Register L3 (PACRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11MD[2:0]			-	PA10MD[2:0]			-	PA9MD[2:0]			-	PA8MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA11MD[2:0]	000*	R/W	PA11 Mode Select the function of the PA11/$\overline{CS1}$/IRQ1/TIC5V/CRx0/RXD0/TX_EN pin. 000: PA11 I/O (port) 001: $\overline{CS1}$ output (BSC) 010: Setting prohibited 011: IRQ1 input (INTC) 100: TIC5V input (MTU2) 101: CRx0 input (RCAN-ET) 110: RXD0 input (SCI) 111: TX_EN input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA10MD[2:0]	000*	R/W	PA10 Mode Select the function of the PA10/$\overline{CS2}$/IRQ2/TIC5W/CTx0/TXD0/MII_TXD0 pin. 000: PA10 I/O (port) 001: $\overline{CS2}$ output (BSC) 010: Setting prohibited 011: IRQ2 input (INTC) 100: TIC5W input (MTU2) 101: CTx0 output (RCAN-ET) 110: TXD0 output (SCI) 111: MII_TXD0 output (Ether)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PA9MD[2:0]	000	R/W	PA9 Mode Select the function of the PA9/$\overline{CS3}$/IRQ3/TCLKD/SSLO/SCK0/MII_TXD1 pin. 000: PA9 I/O (port) 001: $\overline{CS3}$ output (BSC) 010: Setting prohibited 011: IRQ3 input (INTC) 100: TCLKD input (MTU2) 101: SSLO I/O (RSPI) 110: SCK0 I/O (SCI) 111: MII_TXD1 output (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PA8MD[2:0]	000	R/W	PA8 Mode Select the function of the PA8/$\overline{CS4}$/IRQ4/TCLKC/MISO/RXD1/MII_TXD2 pin. 000: PA8 I/O (port) 001: $\overline{CS4}$ output (BSC) 010: Setting prohibited 011: IRQ4 input (INTC) 100: TCLKC input (MTU2) 101: MISO I/O (RSPI) 110: RXD1 input (SCI) 111: MII_TXD2 output (Ether)

- Port A Control Register L2 (PACRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7MD[2:0]			-	PA6MD[2:0]			-	PA5MD[2:0]			-	PA4MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA7MD[2:0]	000	R/W	PA7 Mode Select the function of the PA7/$\overline{CS5}$/IRQ5/TCLKB/MOSI/TXD1/MII_TXD3 pin. 000: PA7 I/O (port) 001: $\overline{CS5}$ output (BSC) 010: Setting prohibited 011: IRQ5 input (INTC) 100: TCLKB input (MTU2) 101: MOSI I/O (RSPI) 110: TXD1 output (SCI) 111: MII_TXD3 output (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA6MD[2:0]	000	R/W	PA6 Mode Select the function of the PA6/$\overline{CS6}$/IRQ6/TCLKA/RSPCK/SCK1/TX_ER pin. 000: PA6 I/O (port) 001: $\overline{CS6}$ output (BSC) 010: Setting prohibited 011: IRQ6 input (INTC) 100: TCLKA input (MTU2) 101: RSPCK I/O (RSPI) 110: SCK1 I/O (SCI) 111: TX_ER output (Ether)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PA5MD[2:0]	000	R/W	PA5 Mode Select the function of the PA5/$\overline{CS5}$/TCLKA/RSPCK/SCK1/RX_ER pin. 000: PA5 I/O (port) 001: $\overline{CS5}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TCLKA input (MTU2) 101: RSPCK I/O (RSPI) 110: SCK1 I/O (SCI) 111: RX_ER output (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PA4MD[2:0]	000	R/W	PA4 Mode Select the function of the PA4/$\overline{CS4}$/TCLKB/MOSI/TXD1/MII_RXD3 pin. 000: PA4 I/O (port) 001: $\overline{CS4}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TCLKB input (MTU2) 101: MOSI I/O (RSPI) 110: TXD1 output (SCI) 111: MII_RXD3 input (Ether)

- Port A Control Register L1 (PACRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3MD[2:0]			-	PA2MD[2:0]			-	PA1MD[2:0]			-	PA0MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA3MD[2:0]	000	R/W	PA3 Mode Select the function of the PA3/$\overline{CS3}$/TCLKC/MISO/RXD1/MII_RXD2 pin. 000: PA3 I/O (port) 001: $\overline{CS3}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TCLKC input (MTU2) 101: MISO I/O (RSPI) 110: RXD1 input (SCI) 111: MII_RXD2 input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA2MD[2:0]	000	R/W	PA2 Mode Select the function of the PA2/$\overline{CS2}$/TCLKD/SSLO/SCK0/MII_RXD1 pin. 000: PA2 I/O (port) 001: $\overline{CS2}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TCLKD input (MTU2) 101: SSLO I/O (RSPI) 110: SCK0 I/O (SCI) 111: MII_RXD1 input (Ether)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PA1MD[2:0]	000	R/W	PA1 Mode Select the function of the PA1/$\overline{CS1}$/IRQ5/CTx0/TXD0/MII_RXD0 pin. 000: PA1 I/O (port) 001: $\overline{CS1}$ output (BSC) 010: Setting prohibited 011: IRQ5 input (INTC) 100: Setting prohibited 101: CTx0 output (RCAN-ET) 110: TXD0 output (SCI) 111: MII_RXD0 input (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PA0MD[2:0]	000	R/W	PA0 Mode Select the function of the PA0/$\overline{CS0}$/IRQ4/CRx0/RXD0/RX_CLK pin. 000: PA0 I/O (port) 001: $\overline{CS0}$ output (BSC) 010: Setting prohibited 011: IRQ4 input (INTC) 100: Setting prohibited 101: CRx0 input (RCAN-ET) 110: RXD0 input (SCI) 111: RX_CLK input (Ether)

22.1.3 Port A Pull-Up MOS Control Registers H and L (PAPCRH and PAPCRL)

PAPCRH and PAPCRL control on and off of the input pull-up MOS of port A in bits.

- Port A Pull-Up MOS Control Register H (PAPCRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA21 PCR	PA20 PCR	PA19 PCR	PA18 PCR	PA17 PCR	PA16 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PA21PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
4	PA20PCR	0	R/W	
3	PA19PCR	0	R/W	
2	PA18PCR	0	R/W	
1	PA17PCR	0	R/W	
0	PA16PCR	0	R/W	

- Port A Pull-Up MOS Control Register L (PAPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PCR	PA14 PCR	PA13 PCR	PA12 PCR	PA11 PCR	PA10 PCR	PA9 PCR	PA8 PCR	PA7 PCR	PA6 PCR	PA5 PCR	PA4 PCR	PA3 PCR	PA2 PCR	PA1 PCR	PA0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PA14PCR	0	R/W	
13	PA13PCR	0	R/W	
12	PA12PCR	0	R/W	
11	PA11PCR	0	R/W	
10	PA10PCR	0	R/W	
9	PA9PCR	0	R/W	
8	PA8PCR	0	R/W	
7	PA7PCR	0	R/W	
6	PA6PCR	0	R/W	
5	PA5PCR	0	R/W	
4	PA4PCR	0	R/W	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

22.1.4 Port B I/O Register L (PBIORL)

PBIORL is a 16-bit readable/writable register that is used to set the pins on port B as inputs or outputs. Bits PB15IOR to PB0IOR correspond to pins PB15 to PB0, respectively (multiplexed port pin names except for the port names are abbreviated here). PBIORL is enabled when the port B pins are functioning as general-purpose inputs/outputs (PB15 to PB0 for PBIORL) or TIOC input/output for the MTU2. In other states, PBIORL is disabled. A given pin on port B will be an output pin if the corresponding bit in PBIORL is set to 1, and an input pin if the bit is cleared to 0. However, settings for bits 13 and 12 in PBIORL are invalid.

The initial value of PBIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15 IOR	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.5 Port B Control Registers L1 to L4 (PBCRL1 to PBCRL4)

PBCRL1 to PBCRL4 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port B.

- Port B Control Register L4 (PBCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB15MD[2:0]			-	PB14MD[2:0]			-	PB13MD[2:0]			-	PB12MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PB15MD[2:0]	000	R/W	PB15 Mode Select the function of the PB15/IRQ7 pin. 000: PB15 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ7 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is read as 0. The write value should always be 0.
10 to 8	PB14MD[2:0]	000	R/W	PB14 Mode Select the function of the PB14/IRQ6 pin. 000: PB14 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ6 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PB13MD[2:0]	000	R/W	PB13 Mode Select the function of the PB13/IRQ3/$\overline{\text{POE2}}$/SDA pin. 000: PB13 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ3 input (INTC) 100: Setting prohibited 101: $\overline{\text{POE2}}$ input (POE2) 110: SDA I/O (IIC3) 111: Setting prohibited
3	—	0	R	Reserved This bit is read as 0. The write value should always be 0.
2 to 0	PB12MD[2:0]	000	R/W	PB12 Mode Select the function of the PB12/IRQ2/$\overline{\text{POE1}}$/SCL pin. 000: PB12 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ2 input (INTC) 100: Setting prohibited 101: $\overline{\text{POE1}}$ input (POE2) 110: SCL I/O (IIC3) 111: Setting prohibited

- Port B Control Register L3 (PBCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB11MD[2:0]			-	PB10MD[2:0]			-	PB9MD[2:0]			-	PB8MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PB11MD[2:0]	000*	R/W	PB11 Mode Select the function of the PB11/$\overline{CS1}$/$\overline{CS3}$/IRQ1/TXD2/$\overline{CS7}$ pin. 000: PB11 I/O (port) 001: $\overline{CS1}$ output (BSC) 010: $\overline{CS3}$ output (BSC) 011: IRQ1 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: TXD2 output (SCI) 111: $\overline{CS7}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB10MD[2:0]	000*	R/W	PB10 Mode Select the function of the PB10/$\overline{CS0}$/$\overline{CS2}$/IRQ0/RXD2/$\overline{CS6}$ pin. 000: PB10 I/O (port) 001: $\overline{CS0}$ output (BSC) 010: $\overline{CS2}$ output (BSC) 011: IRQ0 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: RXD2 input (SCI) 111: $\overline{CS6}$ output (BSC)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PB9MD[2:0]	000*	R/W	PB9 Mode Select the function of the PB9/A25/DACK0/TCLKA/TXD4/$\overline{CS3}$ pin. 000: PB9 I/O (port) 001: A25 output (BSC) 010: DACK0 output (BSC) 011: Setting prohibited 100: TCLKA input (MTU2) 101: Setting prohibited 110: TXD4 output (SCI) 111: $\overline{CS3}$ output (BSC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PB8MD[2:0]	000*	R/W	PB8 Mode Select the function of the PB8/A24/DREQ0/TCLKB/RXD4/$\overline{CS2}$ pin. 000: PB8 I/O (port) 001: A24 output (BSC) 010: DREQ0 input (DMAC) 011: Setting prohibited 100: TCLKB input (MTU2) 101: Setting prohibited 110: RXD4 input (SCI) 111: $\overline{CS2}$ output (BSC)

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB7MD[2:0]			-	PB6MD[2:0]			-	PB5MD[2:0]			-	PB4MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PB7MD[2:0]	000*	R/W	PB7 Mode Select the function of the PB7/A23/TEND0/IRQ7/TCLKC/SCK4/RD/$\overline{WR}$ pin. 000: PB7 I/O (port) 001: A23 output (BSC) 010: TEND0 output (DMAC) 011: IRQ7 input (INTC) 100: TCLKC input (MTU2) 101: Setting prohibited 110: SCK4 I/O (SCI) 111: RD/$\overline{WR}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB6MD[2:0]	000*	R/W	PB6 Mode Select the function of the PB6/A22/$\overline{WAIT}$/IRQ6/TCLKD/TXD0 pin. 000: PB6 I/O (port) 001: A22 output (BSC) 010: $\overline{WAIT}$ input (BSC) 011: IRQ6 input (INTC) 100: TCLKD input (MTU2) 101: Setting prohibited 110: TXD0 output (SCI) 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PB5MD[2:0]	000*	R/W	PB5 Mode Select the function of the PB5/A21/$\overline{\text{BREQ}}$/IRQ5/RXD0 pin. 000: PB5 I/O (port) 001: A21 output (BSC) 010: $\overline{\text{BREQ}}$ input (BSC) 011: IRQ5 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: RXD0 input (SCI) 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PB4MD[2:0]	000*	R/W	PB4 Mode Select the function of the PB4/A20/$\overline{\text{BACK}}$/IRQ4/TIOC0D/WAIT/SCK3/$\overline{\text{BS}}$ pin. 000: PB4 I/O (port) 001: A20 output (BSC) 010: $\overline{\text{BACK}}$ output (BSC) 011: IRQ4 input (INTC) 100: TIOC0D I/O (MTU2) 101: WAIT input (BSC) 110: SCK3 I/O (SCI) 111: $\overline{\text{BS}}$ input (BSC)

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port B Control Register L1 (PBCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB3MD[2:0]			-	PB2MD[2:0]			-	PB1MD[2:0]			-	PB0MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PB3MD[2:0]	000*[1]	R/W	PB3 Mode Select the function of the PB3/A19/$\overline{\text{BREQ}}$/IRQ3/TIOC0C/$\overline{\text{CASL}}$/TXD3/$\overline{\text{AH}}$ pin. 000: PB3 I/O (port) 001: A19 output (BSC) 010: $\overline{\text{BREQ}}$ input (BSC) 011: IRQ3 input (INTC) 100: TIOC0C I/O (MTU2) 101: $\overline{\text{CASL}}$ output (BSC) 110: TXD3 output (SCI) 111: $\overline{\text{AH}}$ output (BSC)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB2MD[2:0]	000*[1]	R/W	PB2 Mode Select the function of the PB2/A18/$\overline{\text{BACK}}$/IRQ2/TIOC0B/$\overline{\text{RASL}}$/RXD3/$\overline{\text{FRAME}}$ pin. 000: PB2 I/O (port) 001: A18 output (BSC) 010: $\overline{\text{BACK}}$ output (BSC) 011: IRQ2 input (INTC) 100: TIOC0B I/O (MTU2) 101: $\overline{\text{RASL}}$ output (BSC) 110: RXD3 input (SCI) 111: $\overline{\text{FRAME}}$ output (BSC)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PB1MD[2:0]	000*[1]	R/W	PB1 Mode Select the function of the PB1/A17/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/IRQ1/TIOC0A/$\overline{\text{ADTRG}}$ pin. 000: PB1 I/O (port) 001: A17/output (BSC) 010: $\overline{\text{IRQOUT}}$ output (INTC)/$\overline{\text{REFOUT}}$ output (BSC)*[2] 011: IRQ1 input (INTC) 100: TIOC0A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: $\overline{\text{ADTRG}}$ input (ADC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PB0MD[2:0]	000*[1]	R/W	PB0 Mode Select the function of the PB0/A16/RD/$\overline{\text{WR}}$/IRQ0/TIOC2A pin. 000: PB0 I/O (port) 001: A16 output (BSC) 010: RD/$\overline{\text{WR}}$ output (BSC) 011: IRQ0 input (INTC) 100: TIOC2A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Notes: 1. The initial value is 001 during the on-chip ROM disabled external extension mode.
2. Setting of the IRQOUT function control register (IFCR) selects $\overline{\text{IRQOUT}}$ (INTC) or $\overline{\text{REFOUT}}$ (BSC).

22.1.6 Port B Pull-Up MOS Control Register L (PBPCRL)

PBPCRL controls on/off of the input pull-up MOS of port B in bits.

- Port B Pull-Up MOS Control Register L (PBPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15 PCR	PB14 PCR	PB13 PCR	PB12 PCR	PB11 PCR	PB10 PCR	PB9 PCR	PB8 PCR	PB7 PCR	PB6 PCR	PB5 PCR	PB4 PCR	PB3 PCR	PB2 PCR	PB1 PCR	PB0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PB15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PB14PCR	0	R/W	
13	PB13PCR	0	R/W	Reserved
12	PB12PCR	0	R/W	The corresponding input pull-up MOS turns on regardless of the setting value.
11	PB11PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
10	PB10PCR	0	R/W	
9	PB9PCR	0	R/W	
8	PB8PCR	0	R/W	
7	PB7PCR	0	R/W	
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

22.1.7 Port C I/O Register L (PCIORL)

PCIORL is a 16-bit readable/writable register that is used to set the pins on port C as inputs or outputs. Bits PC15IOR to PC0IOR correspond to pins PC15 to PC0, respectively (multiplexed port pin names except for the port names are abbreviated here). PCIORL is enabled when the port C pins are functioning as general-purpose inputs/outputs (PC15 to PC0). In other states, PCIORL is disabled. A given pin on port C will be an output pin if the corresponding bit in PCIORL is set to 1, and an input pin if the bit is cleared to 0. The initial value of PCIORL is H'0000.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.8 Port C Control Registers L1 to L4 (PCCRL1 to PCCRL4)

PCCRL1 to PACRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port C.

- Port C Control Register L4 (PCCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC15MD[2:0]			-	PC14MD[2:0]			-	PC13MD[2:0]			-	PC12MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PC15MD[2:0]	000*	R/W	PC15 Mode Select the function of the PC15/A15/IRQ2/TCLKD pin. 000: PC15 I/O (port) 001: A15 output (BSC) 010: Setting prohibited 011: IRQ2 input (INTC) 100: TCLKD input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PC14MD[2:0]	000*	R/W	PC14 Mode Select the function of the PC14/A14/IRQ1/TCLKC pin. 000: PC14 I/O (port) 001: A14 output (BSC) 010: Setting prohibited 011: IRQ1 input (INTC) 100: TCLKC input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PC13MD[2:0]	000*	R/W	PC13 Mode Select the function of the PC13/A13/IRQ0/TCLKB pin. 000: PC13 I/O (port) 001: A13 output (BSC) 010: Setting prohibited 011: IRQ0 input (INTC) 100: TCLKB input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PC12MD[2:0]	000*	R/W	PC12 Mode Select the function of the PC12/A12/TCLKA pin. 000: PC12 I/O (port) 001: A12 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TCLKA input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port C Control Register L3 (PCCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC11MD[2:0]			-	PC10MD[2:0]			-	PC9MD[2:0]			-	PC8MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PC11MD[2:0]	000*	R/W	PC11 Mode Select the function of the PC11/A11/TIOC1B/CTx0/TXD0 pin. 000: PC11 I/O (port) 001: A11 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC1B I/O (MTU2) 101: CTx0 output (RCAN-ET) 110: TXD0 input (SCI) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PC10MD[2:0]	000*	R/W	PC10 Mode Select the function of the PC10/A10/TIOC1A/CRx0/RXD0 pin. 000: PC10 I/O (port) 001: A10 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC1A I/O (MTU2) 101: CRx0 input (RCAN-ET) 110: RXD0 input (SCI) 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PC9MD[2:0]	000*	R/W	PC9 Mode Select the function of the PC9/A9/CTx0/TXD0 pin. 000: PC9 I/O (port) 001: A9 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: CTx0 output (RCAN-ET) 110: TXD0 output (SCI) 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PC8MD[2:0]	000*	R/W	PC8 Mode Select the function of the PC8/A8/CRx0/RXD0 pin. 000: PC8 I/O (port) 001: A8 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: CRx0 input (RCAN-ET) 110: RXD0 input (SCI) 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port C Control Register L2 (PCCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC7MD[2:0]			-	PC6MD[2:0]			-	PC5MD[2:0]			-	PC4MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PC7MD[2:0]	000*	R/W	PC7 Mode Select the function of the PC7/A7 pin. 000: PC7 I/O (port) 001: A7 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PC6MD[2:0]	000*	R/W	PC6 Mode Select the function of the PC6/A6 pin. 000: PC6 I/O (port) 001: A6 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PC5MD[2:0]	000*	R/W	PC5 Mode Select the function of the PC5/A5 pin. 000: PC5 I/O (port) 001: A5 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PC4MD[2:0]	000*	R/W	PC4 Mode Select the function of the PC4/A4 pin. 000: PC4 I/O (port) 001: A4 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port C Control Register L1 (PCCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC3MD[2:0]			-	PC2MD[2:0]			-	PC1MD[2:0]			-	PC0MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PC3MD[2:0]	000*	R/W	PC3 Mode Select the function of the PC3/A3 pin. 000: PC3 I/O (port) 001: A3 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PC2MD[2:0]	000*	R/W	PC2 Mode Select the function of the PC2/A2 pin. 000: PC2 I/O (port) 001: A2 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PC1MD[2:0]	000*	R/W	PC1 Mode Select the function of the PC1/A1 pin. 000: PC1 I/O (port) 001: A1 output (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PC0MD[2:0]	000*	R/W	PC0 Mode Select the function of the PC0/A0/IRQ4/$\overline{\text{POE0}}$ pin. 000: PC0 I/O (port) 001: A0 output (BSC) 010: Setting prohibited 011: IRQ4 input (INTC) 100: Setting prohibited 101: $\overline{\text{POE0}}$ input (POE2) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

22.1.9 Port C Pull-Up MOS Control Register L (PCPCRL)

PCPCRL controls on/off of the input pull-up MOS of port C in bits.

- Port C Pull-Up MOS Control Register L (PCPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 PCR	PC14 PCR	PC13 PCR	PC12 PCR	PC11 PCR	PC10 PCR	PC9 PCR	PC8 PCR	PC7 PCR	PC6 PCR	PC5 PCR	PC4 PCR	PC3 PCR	PC2 PCR	PC1 PCR	PC0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PC15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PC14PCR	0	R/W	
13	PC13PCR	0	R/W	
12	PC12PCR	0	R/W	
11	PC11PCR	0	R/W	
10	PC10PCR	0	R/W	
9	PC9PCR	0	R/W	
8	PC8PCR	0	R/W	
7	PC7PCR	0	R/W	
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

22.1.10 Port D I/O Registers H and L (PDIORH and PDIORL)

PDIORH and PDIORL are 16-bit readable/writable registers that are used to set the pins on port D as inputs or outputs. Bits PD31IOR to PD0IOR correspond to pins PD31 to PD0, respectively (multiplexed port pin names except for the port names are abbreviated here). PDIORL is enabled when the port D pins are functioning as general-purpose inputs/outputs (PD15 to PD0 for PDIORL) and TIOC inputs/outputs in MTU2S. In other states, PDIORL is disabled. PDIORH is enabled when the port D pins are functioning as general-purpose inputs/outputs (PD31 to PD16 for PDIORH) and TIOC inputs/outputs in MTU2S. In other states, PDIORH is disabled. A given pin on port D will be an output pin if the corresponding bit in PDIORL and PDIORH is set to 1, and an input pin if the bit is cleared to 0.

The initial values of PDIORL and PDIORH are both H'0000.

- Port D I/O Register H (PDIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 IOR	PD30 IOR	PD29 IOR	PD28 IOR	PD27 IOR	PD26 IOR	PD25 IOR	PD24 IOR	PD23 IOR	PD22 IOR	PD21 IOR	PD20 IOR	PD19 IOR	PD18 IOR	PD17 IOR	PD16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Port D I/O Register L (PDIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.11 Port D Control Registers H1 to H4 and L1 to L4 (PDCRH1 to PDCRH4 and PDCRL1 to PDCRL4)

PDCRH1 to PDCRH4 and PDCRL1 to PDCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port D.

- Port D Control Register H4 (PDCRH4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD31MD[2:0]			-	PD30MD[2:0]			-	PD29MD[2:0]			-	PD28MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD31MD[2:0]	000*	R/W	PD31 Mode Select the function of the PD31/D31/TIOC3AS/SSL2/RX_DV pin. 000: PD31 I/O (port) 001: D31 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3AS I/O (MTU2S) 110: SSL2 output (RSPI) 111: RX_DV input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PD30MD[2:0]	000*	R/W	PD30 Mode Select the function of the PD30/D30/TIOC3CS/SSL3/RX_ER pin. 000: PD30 I/O (port) 001: D30 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3C I/O (MTU2S) 110: SSL3 output (RSPI) 111: RX_ER input (Ether)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD29MD[2:0]	000*	R/W	PD29 Mode Select the function of the PD29/D29/TIOC3BS/MII_RXD3 pin. 000: PD29 I/O (port) 001: D29 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3BS I/O (MTU2S) 110: Setting prohibited 111: MII_RXD3 input (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PD28MD[2:0]	000*	R/W	PD28 Mode Select the function of the PD28/D28/TIOC3DS/MII_RXD2 pin. 000: PD28 I/O (port) 001: D28 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3DS I/O (MTU2S) 110: Setting prohibited 111: MII_RXD2 input (Ether)

Note: * The initial value is 001 during the on-chip ROM disabled 32-bit external extension mode.

- Port D Control Register H3 (PDCRH3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD27MD[2:0]			-	PD26MD[2:0]			-	PD25MD[2:0]			-	PD24MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PD27MD[2:0]	000*	R/W	PD27 Mode Select the function of the PD27/D27/TIOC4AS/MII_RXD1 pin. 000: PD27 I/O (port) 001: D27 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4AS I/O (MTU2S) 110: Setting prohibited 111: MII_RXD1 input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD26MD[2:0]	000*	R/W	PD26 Mode Select the function of the PD26/D26/TIOC4BS/MII_RXD0 pin. 000: PD26 I/O (port) 001: D26 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4BS I/O (MTU2S) 110: Setting prohibited 111: MII_RXD0 input (Ether)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PD25MD[2:0]	000*	R/W	PD25 Mode Select the function of the PD25/D25/TIOC4CS/RX_CLK pin. 000: PD25 I/O (port) 001: D25 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4CS I/O (MTU2S) 110: Setting prohibited 111: RX_CLK input (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD24MD[2:0]	000*	R/W	PD24 Mode Select the function of the PD24/D24/TIOC4DS/CRS pin. 000: PD24 I/O (port) 001: D24 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4DS I/O (MTU2S) 110: Setting prohibited 111: CRS input (Ether)

Note: * The initial value is 001 during the on-chip ROM disabled 32-bit external extension mode.

- Port D Control Register H2 (PDCRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD23MD[2:0]			-	PD22MD[2:0]			-	PD21MD[2:0]			-	PD20MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD23MD[2:0]	000*	R/W	PD23 Mode Select the function of the PD23/D23/DACK1/IRQ7/COL pin. 000: PD23 I/O (port) 001: D23 I/O (BSC) 010: DACK1 output (DMAC) 011: IRQ7 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: COL input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD22MD[2:0]	000*	R/W	PD22 Mode Select the function of the PD22/D22/DREQ1/IRQ6/WOL pin. 000: PD22 I/O (port) 001: D22 I/O (BSC) 010: DREQ1 input (DMAC) 011: IRQ6 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: WOL output (Ether)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD21MD[2:0]	000*	R/W	PD21 Mode Select the function of the PD21/D21/TEND1/IRQ5/AUDCK/EXOUT pin. 000: PD21 I/O (port) 001: D21 I/O (BSC) 010: TEND1 output (DMAC) 011: IRQ5 input (INTC) 100: AUDCK output (AUD) 101: Setting prohibited 110: Setting prohibited 111: EXOUT output (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD20MD[2:0]	000*	R/W	PD20 Mode Select the function of the PD20/D20/IRQ4/$\overline{\text{AUDSYNC}}$/MDC pin. 000: PD20 I/O (port) 001: D20 I/O (BSC) 010: Setting prohibited 011: IRQ4 input (INTC) 100: $\overline{\text{AUDSYNC}}$ output (AUD) 101: Setting prohibited 110: Setting prohibited 111: MDC output (Ether)

Note: * The initial value is 001 during the on-chip ROM disabled 32-bit external extension mode.

- Port D Control Register H1 (PDCRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD19MD[2:0]			-	PD18MD[2:0]			-	PD17MD[2:0]			-	PD16MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled 32-bit external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD19MD[2:0]	000*	R/W	PD19 Mode Select the function of the PD19/D19/IRQ3/AUDATA3/LNKSTA pin. 000: PD19 I/O (port) 001: D19 I/O (port) 010: Setting prohibited 011: IRQ3 input (INTC) 100: AUDATA3 output (AUD) 101: Setting prohibited 110: Setting prohibited 111: LNKSTA input (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD18MD[2:0]	000*	R/W	PD18 Mode Select the function of the PD18/D18/IRQ2/AUDATA2/MDIO pin. 000: PD18 I/O (port) 001: D18 I/O (BSC) 010: Setting prohibited 011: IRQ2 input (INTC) 100: AUDATA2 output (AUD) 101: Setting prohibited 110: Setting prohibited 111: MDIO I/O (Ether)

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD17MD[2:0]	000*	R/W	PD17 Mode Select the function of the PD17/D17/IRQ1/AUDATA1/$\overline{\text{POE4}}$/$\overline{\text{ADTRG}}$ pin. 000: PD17 I/O (port) 001: D17 I/O (BSC) 010: Setting prohibited 011: IRQ1 input (INTC) 100: AUDATA1 output (AUD) 101: $\overline{\text{POE4}}$ input (POE2) 110: Setting prohibited 111: $\overline{\text{ADTRG}}$ input (ADC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD16MD[2:0]	000*	R/W	PD16 Mode Select the function of the PD16/D16/$\overline{\text{UBCTRG}}$/IRQ0/AUDATA0/$\overline{\text{POE0}}$ pin. 000: PD16 I/O (port) 001: D16 I/O (BSC) 010: $\overline{\text{UBCTRG}}$ output (UBC) 011: IRQ0 input (INTC) 100: AUDATA0 output (AUD) 101: $\overline{\text{POE0}}$ input (POE2) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled 32-bit external extension mode.

- Port D Control Register L4 (PDCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD15MD[2:0]			-	PD14MD[2:0]			-	PD13MD[2:0]			-	PD12MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD15MD[2:0]	000*	R/W	PD15 Mode Select the function of the PD15/D15/TIOC4DS pin. 000: PD15 I/O (port) 001: D15 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4DS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD14MD[2:0]	000*	R/W	PD14 Mode Select the function of the PD14/D14/TIOC4CS pin. 000: PD14 I/O (port) 001: D14 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4CS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD13MD[2:0]	000*	R/W	PD13 Mode Select the function of the PD13/D13/TIOC4BS pin. 000: PD13 I/O (port) 001: D13 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC4BS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD12MD[2:0]	000*	R/W	PD12 Mode Select the function of the PD12/D12/TIOC4AS pin. 000: PD12 I/O (port) 001: D12 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited) 101: TIOC4AS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port D Control Register L3 (PDCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD11MD[2:0]			-	PD10MD[2:0]			-	PD9MD[2:0]			-	PD8MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD11MD[2:0]	000*	R/W	PD11 Mode Select the function of the PD11/D11/TIOC3DS pin. 000: PD11 I/O (port) 001: D11 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3DS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD10MD[2:0]	000*	R/W	PD10 Mode Select the function of the PD10/D10/TIOC3BS pin. 000: PD10 I/O (port) 001: D10 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3BS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD9MD[2:0]	000*	R/W	PD9 Mode Select the function of the PD9/D9/TIOC3CS pin. 000: PD9 I/O (port) 001: D9 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3CS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD8MD[2:0]	000*	R/W	PD8 Mode Select the function of the PD8/D8/TIOC3AS pin. 000: PD8 I/O (port) 001: D8 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIOC3AS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port D Control Register L2 (PDCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD7MD[2:0]			-	PD6MD[2:0]			-	PD5MD[2:0]			-	PD4MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD7MD[2:0]	000*	R/W	PD7 Mode Select the function of the PD7/D7/TIC5WS pin. 000: PD7 I/O (port) 001: D7 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIC5WS input (MTU2S) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD6MD[2:0]	000*	R/W	PD6 Mode Select the function of the PD6/D6/TIC5VS pin. 000: PD6 I/O (port) 001: D6 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIC5VS input (MTU2S) 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD5MD[2:0]	000*	R/W	PD5 Mode Select the function of the PD5/D5/TIC5US pin. 000: PD5 I/O (port) 001: D5 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TIC5US input (MTU2S) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD4MD[2:0]	000*	R/W	PD4 Mode Select the function of the PD4/D4/TIC5W/SCK2 pin. 000: PD4 I/O (port) 001: D4 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIC5W input (MTU2) 101: Setting prohibited 110: SCK2 I/O (SCI) 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

- Port D Control Register L1 (PDCRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD3MD[2:0]			-	PD2MD[2:0]			-	PD1MD[2:0]			-	PD0MD[2:0]		
Initial value:	0	0	0	0*	0	0	0	0*	0	0	0	0*	0	0	0	0*
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value is 1 during the on-chip ROM disabled external extension mode.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PD3MD[2:0]	000*	R/W	PD3 Mode Select the function of the PD3/D3/TIC5V/TXD2 pin. 000: PD3 I/O (port) 001: D3 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIC5V input (MTU2) 101: Setting prohibited 110: TXD2 output (SCI) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PD2MD[2:0]	000*	R/W	PD2 Mode Select the function of the PD2/D2/TIC5U/RXD2 pin. 000: PD2 I/O (port) 001: D2 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIC5U input (MTU2) 101: Setting prohibited 110: RXD2 input (SCI) 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PD1MD[2:0]	000*	R/W	PD1 Mode Select the function of the PD1/D1 pin. 000: PD1 I/O (port) 001: D1 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD0MD[2:0]	000*	R/W	PD0 Mode Select the function of the PD0/D0 pin. 000: PD0 I/O (port) 001: D0 I/O (BSC) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value is 001 during the on-chip ROM disabled external extension mode.

22.1.12 Port D Pull-Up MOS Control Registers H and L (PDPCRH and PDPCRL)

PDPCRH and PDPCRL control on/off of the input pull-up MOS of port D in bits.

- Port D Pull-Up MOS Control Register H (PDPCRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 PCR	PD30 PCR	PD29 PCR	PD28 PCR	PD27 PCR	PD26 PCR	PD25 PCR	PD24 PCR	PD23 PCR	PD22 PCR	PD21 PCR	PD20 PCR	PD19 PCR	PD18 PCR	PD17 PCR	PD16 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD31PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PD30PCR	0	R/W	
13	PD29PCR	0	R/W	
12	PD28PCR	0	R/W	
11	PD27PCR	0	R/W	
10	PD26PCR	0	R/W	
9	PD25PCR	0	R/W	
8	PD24PCR	0	R/W	
7	PD23PCR	0	R/W	
6	PD22PCR	0	R/W	
5	PD21PCR	0	R/W	
4	PD20PCR	0	R/W	
3	PD19PCR	0	R/W	
2	PD18PCR	0	R/W	
1	PD17PCR	0	R/W	
0	PD16PCR	0	R/W	

- Port D Pull-Up MOS Control Register L (PDPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 PCR	PD14 PCR	PD13 PCR	PD12 PCR	PD11 PCR	PD10 PCR	PD9 PCR	PD8 PCR	PD7 PCR	PD6 PCR	PD5 PCR	PD4 PCR	PD3 PCR	PD2 PCR	PD1 PCR	PD0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PD14PCR	0	R/W	
13	PD13PCR	0	R/W	
12	PD12PCR	0	R/W	
11	PD11PCR	0	R/W	
10	PD10PCR	0	R/W	
9	PD9PCR	0	R/W	
8	PD8PCR	0	R/W	
7	PD7PCR	0	R/W	
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

22.1.13 Port E I/O Register L (PEIORL)

PEIORL is a 16-bit readable/writable register that is used to set the pins on port E as inputs or outputs. Bits PE15IOR to PE0IOR correspond to pins PE15 to PE0, respectively (multiplexed port pin names except for the port names are abbreviated here). PEIORL is enabled when the port E pins are functioning as general-purpose inputs/outputs (PE15 to PE0 for PEIORL) and TIOC inputs/outputs in both MTU2 and MTU2S. In other states, PEIORL is disabled. A given pin on port E will be an output pin if the corresponding bit in PEIORL is set to 1, and an input pin if the bit is cleared to 0.

The initial value of PEIORL is H'0000.

- Port E I/O Register L (PEIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.1.14 Port E Control Registers L1 to L4 (PECRL1 to PECRL4)

PECRL1 to PECRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port E.

- Port E Control Register L4 (PECRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15MD[2:0]			-	PE14MD[2:0]			-	PE13MD[2:0]			-	PE12MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PE15MD[2:0]	000	R/W	PE15 Mode Select the function of the PE15/DACK1/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/TIOC4D/TX_ER pin. 000: PE15 I/O (port) 001: Setting prohibited 010: DACK1 output (DMAC) 011: $\overline{\text{IRQOUT}}$ output (INTC)/$\overline{\text{REFOUT}}$ output (BSC)* 100: TIOC4D I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: TX_ER output (Ether)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PE14MD[2:0]	000	R/W	PE14 Mode Select the function of the PE14/DACK0/TIOC4C/MII_TXD3 pin. 000: PE14 I/O (port) 001: Setting prohibited 010: DACK0 output (DMAC) 011: Setting prohibited 100: TIOC4C I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: MII_TXD3 output (Ether)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PE13MD[2:0]	000	R/W	PE13 Mode Select the function of the PE13/$\overline{\text{MRES}}$/TIOC4B/MII_TXD2 pin. 000: PE13 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: $\overline{\text{MRES}}$ input (system control) 100: TIOC4B I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: MII_TXD2 output (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PE12MD[2:0]	000	R/W	PE12 Mode Select the function of the PE12/TIOC4A/MII_TXD1 pin. 000: PE12 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: TIOC4A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: MII_TXD1 output (Ether)

- Port E Control Register L3 (PECRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE11MD[2:0]			-	PE10MD[2:0]			-	PE9MD[2:0]			-	PE8MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PE11MD[2:0]	000	R/W	PE11 Mode Select the function of the PE11/DACK3/TIOC3D/MII_TXD pin. 000: PE11 I/O (port) 001: Setting prohibited 010: DACK3 output (DMAC) 011: Setting prohibited 100: TIOC3D I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: MII_TXD output (Ether)

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PE10MD[2:0]	000	R/W	PE10 Mode Select the function of the PE10/DREQ3/TIOC3C/SSL3/TXD2/TX_CLK pin. 000: PE10 I/O (port) 001: Setting prohibited 010: DREQ3 input (DMAC) 011: Setting prohibited 100: TIOC3C I/O (MTU2) 101: SSL3 output (RSPI) 110: TXD2 output (SCI) 111: TX_CLK input (Ether)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PE9MD[2:0]	000	R/W	PE9 Mode Select the function of the PE9/DACK2/TIOC3B/TX_EN pin. 000: PE9 I/O (port) 001: Setting prohibited 010: DACK2 output (DMAC) 011: Setting prohibited 100: TIOC3B I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: TX_EN output (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PE8MD[2:0]	000	R/W	PE8 Mode Select the function of the PE8/DREQ2/TIOC3A/SSL2/SCK2/EXOUT pin. 000: PE8 I/O (port) 001: Setting prohibited 010: DREQ2 input (DMAC) 011: Setting prohibited 100: TIOC3A I/O (MTU2) 101: SSL2 output (RSPI) 110: SCK2 I/O (SCI) 111: EXOUT output (Ether)

- Port E Control Register L2 (PECRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7MD[2:0]			-	PE6MD[2:0]			-	PE5MD[2:0]			-	PE4MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PE7MD[2:0]	000	R/W	PE7 Mode Select the function of the PE7/$\overline{\text{UBCTRG}}$/TIOC2B/SSL1/RXD2/RX_DV pin. 000: PE7 I/O (port) 001: Setting prohibited 010: $\overline{\text{UBCTRG}}$ output (UBC) 011: Setting prohibited 100: TIOC2B I/O (MTU2) 101: SSL1output (RSPI) 110: RXD2 input (SCI) 111: RX_DV input (Ether)

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PE6MD[2:0]	000	R/W	PE6 Mode Select the function of the PE6/TIOC2A/TIOC3DS/RXD3 pin. 000: PE6 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: TIOC2A I/O (MTU2) 101: TIOC3DS I/O (MTU2S) 110: RXD3 input (SCI) 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PE5MD[2:0]	000	R/W	PE5 Mode Select the function of the PE5/TIOC1B/TIOC3BS/TXD3/MDIO pin. 000: PE5 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: TIOC1B I/O (MTU2) 101: TIOC3BS I/O (MTU2S) 110: TXD3 output (SCI) 111: MDIO I/O (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PE4MD[2:0]	000	R/W	PE4 Mode Select the function of the PE4/IRQ4/TIOC1A/$\overline{\text{POE8}}$/SCK3/CRS pin. 000: PE4 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ4 input (INTC) 100: TIOC1A I/O (MTU2) 101: $\overline{\text{POE8}}$ input (POE2) 110: SCK3 I/O (SCI) 111: CRS input (Ether)

- Port E Control Register L1 (PECRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE3MD[2:0]			-	PE2MD[2:0]			-	PE1MD[2:0]			-	PE0MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PE3MD[2:0]	000	R/W	PE3 Mode Select the function of the PE3/TEND1/TIOC0D/TIOC4DS/COL pin. 000: PE3 I/O (port) 001: Setting prohibited 010: TEND1 output (DMAC) 011: Setting prohibited 100: TIOC0D I/O (MTU2) 101: TIOC4DS I/O (MTU2S) 110: Setting prohibited 111: COL input (Ether)

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PE2MD[2:0]	000	R/W	PE2 Mode Select the function of the PE2/DREQ1/TIOC0C/TIOC4CS/WOL pin. 000: PE2 I/O (port) 001: Setting prohibited 010: DREQ1 input (DMAC) 011: Setting prohibited 100: TIOC0C I/O (MTU2) 101: TIOC4CS I/O (MTU2S) 110: Setting prohibited 111: WOL output (Ether)
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6 to 4	PE1MD[2:0]	000	R/W	PE1 Mode Select the function of the PE1/TEND0/TIOC0B/TIOC4BS/MDC pin. 000: PE1 I/O (port) 001: Setting prohibited 010: TEND0 output (DMAC) 011: Setting prohibited 100: TIOC0B I/O (MTU2) 101: TIOC4BS I/O (MTU2S) 110: Setting prohibited 111: MDC output (Ether)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PE0MD[2:0]	000	R/W	PE0 Mode Select the function of the PE0/DREQ0/TIOC0A/TIOC4AS/LNKSTA pin. 000: PE0 I/O (port) 001: Setting prohibited 010: DREQ0 input (DMAC) 011: Setting prohibited 100: TIOC0A I/O (MTU2) 101: TIOC4AS I/O (MTU2S) 110: Setting prohibited 111: LNKSTA input (Ether)

22.1.15 Port E Pull-Up MOS Control Register L (PEPCRL)

PEPCRL controls the on/off of the input pull-up MOS of the port E in bits.

- Port E Pull-Up MOS Control Register L (PEPCRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 PCR	PE14 PCR	PE13 PCR	PE12 PCR	PE11 PCR	PE10 PCR	PE9 PCR	PE8 PCR	PE7 PCR	PE6 PCR	PE5 PCR	PE4 PCR	PE3 PCR	PE2 PCR	PE1 PCR	PE0 PCR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PE15PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	PE14PCR	0	R/W	
13	PE13PCR	0	R/W	
12	PE12PCR	0	R/W	
11	PE11PCR	0	R/W	
10	PE10PCR	0	R/W	
9	PE9PCR	0	R/W	
8	PE8PCR	0	R/W	
7	PE7PCR	0	R/W	
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

22.1.16 Large Current Port Control Register (HCPCR)

HCPCR is a 16-bit readable/writable register that is used to control the large current port. It controls pins PD10 to PD15, PD24 to PD29, PE0 to PE3, PE5, PE6, PE9, and PE11 to PE15.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	MZI ZDH	MZI ZDL	MZI ZEH	MZI ZEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MZIZDH	1	R/W	Port D Large Current Port High Impedance H Selects whether to set the large current port of PD24 to PD29 to the high-impedance state regardless of the setting of the PFC during the oscillation stop detection and software standby mode. 0: set to the high-impedance state 1: do not set to the high-impedance state The pin state is retained during the oscillation stop detection when this bit is set to 1. See appendix A, Pin States, for details on the software standby mode.
2	MZIZDL	1	R/W	Port D Large Current Port High Impedance L Selects whether to set the large current port of PD10 to PD15 to the high-impedance state regardless of the setting of the PFC during the oscillation stop detection and software standby mode. 0: set to the high-impedance state 1: do not set to the high-impedance state The pin state is retained during the oscillation stop detection when this bit is set to 1. See appendix A, Pin States, for details on the software standby mode

Bit	Bit Name	Initial Value	R/W	Description
1	MZIZEH	1	R/W	Port E Large Current Port High Impedance H Selects whether to set the large current port of PE9, and PE11 to PE15 to the high-impedance state regardless of the setting of the PFC during the oscillation stop detection and software standby mode. 0: set to the high-impedance state 1: do not set to the high-impedance state The pin state is retained during the oscillation stop detection when this bit is set to 1. See appendix A, Pin States, for details on the software standby mode
0	MZIZEL	1	R/W	Port E Large Current Port High Impedance L Selects whether to set the large current port of PE0 to PE3, PE5, and PE6 to the high-impedance state regardless of the setting of the PFC during the oscillation stop detection and software standby mode. 0: set to the high-impedance state 1: do not set to the high-impedance state The pin state is retained during the oscillation stop detection when this bit is set to 1. See appendix A, Pin States, for details on the software standby mode.

22.1.17 IRQOUT Function Control Register (IFCR)

IFCR is a 16-bit readable/writable register that is used to control the $\overline{\text{IRQOUT}}$ output and $\overline{\text{REFOUT}}$ output when they are selected as the multiplexed pin functions for PB1 or PE15. If the function selected for the corresponding pin differs from this, the IFCR setting does not affect how the pin functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	IRQ MD3	IRQ MD2	IRQ MD1	IRQ MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	IRQMD3	0	R/W	Port B $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ Pin Function Select
2	IRQMD2	0	R/W	Select $\overline{\text{IRQOUT}}$ or $\overline{\text{REFOUT}}$ as a pin function when bits 6 to 4 (PB1MD2, PB1MD1, and PB1MD0) in PBCRL1 are set to 0, 1, and 0. 00: Interrupt request accept output ($\overline{\text{IRQOUT}}$) 01: Refresh signal output ($\overline{\text{REFOUT}}$) 10: Interrupt request accept output ($\overline{\text{IRQOUT}}$) or refresh signal output ($\overline{\text{REFOUT}}$) (depends on the operating state) 11: Always high-level output
1	IRQMD1	0	R/W	Port E $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ Pin Function Select
0	IRQMD0	0	R/W	Select $\overline{\text{IRQOUT}}$ or $\overline{\text{REFOUT}}$ as a pin function when bits 14 to 12 (PE15MD2, PE15MD1, and PE15MD0) in PECRL4 are set to 0, 1, and 1. 00: Interrupt request accept output ($\overline{\text{IRQOUT}}$) 01: Refresh signal output ($\overline{\text{REFOUT}}$) 10: Interrupt request accept output ($\overline{\text{IRQOUT}}$) or refresh signal output ($\overline{\text{REFOUT}}$) (depends on the operating state) 11: Always high-level output

22.1.18 DACK Output Timing Control Register (PDACKCR)

PDACKCR is a 16-bit readable/writable register that is used to control the timing of the output of signals from the DACK0 to DACK3 pins. If the function selected for the corresponding pin differs from this, the PDACKCR setting does not affect how the pin functions.

Before setting this register, set the AL bit in DMCR to determine the active level for DACK signals. Additionally, when this register is used to change the timing of a DACK output, confirm that this provides the system with enough hold time for the writing of data during single-address transfer.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	DACK3 TMG	DACK2 TMG	DACK1 TMG	DACK0 TMG
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
3	DACK3TMG	0	R/W	DACK3 Pin Timing Select This bit controls timing of the assertion of the DACK3 pin. 0: The intervals over which DACK3 is asserted on the relevant bus interfaces are as indicated below. Normal space: From the beginning of T1 until the end of T2 MPX-I/O: From the beginning of T1 until the end of T2 SRAM with byte selection: From the beginning of Th until the end of Tf Burst ROM: From the beginning of T1 until the end of T2B Synchronous DRAM: From the beginning of Tr until completion of access 1: The intervals over which DACK3 is asserted on the relevant bus interfaces are as indicated below. Normal space: The same as for $\overline{RD}$ or $\overline{WRxx}$ MPX-I/O: The same as for $\overline{RD}$ or $\overline{WRxx}$ Only set this bit to 1 if the area of memory that is the target for transfer at the time of DACK3 assertion is in a normal space or the MPX-I/O space.

Bit	Bit Name	Initial Value	R/W	Description
2	DACK2TMG	0	R/W	DACK2 Pin Timing Select This bit controls timing of the assertion of the DACK2 pin. 0: The intervals over which DACK2 is asserted on the relevant bus interfaces are as indicated below. Normal space: From the beginning of T1 until the end of T2 MPX-I/O: From the beginning of T1 until the end of T2 SRAM with byte selection: From the beginning of Th until the end of Tf Burst ROM: From the beginning of T1 until the end of T2B Synchronous DRAM: From the beginning of Tr until completion of access 1: The intervals over which DACK2 is asserted on the relevant bus interfaces are as indicated below. Normal space: The same as for $\overline{RD}$ or $\overline{WRxx}$ MPX-I/O: The same as for $\overline{RD}$ or $\overline{WRxx}$ Only set this bit to 1 if the area of memory that is the target for transfer at the time of DACK2 assertion is in a normal space or the MPX-I/O space.

Bit	Bit Name	Initial Value	R/W	Description
1	DACK1TMG	0	R/W	DACK1 Pin Timing Select This bit controls timing of the assertion of the DACK1 pin. 0: The intervals over which DACK1 is asserted on the relevant bus interfaces are as indicated below. Normal space: From the beginning of T1 until the end of T2 MPX-I/O: From the beginning of T1 until the end of T2 SRAM with byte selection: From the beginning of Th until the end of Tf Burst ROM: From the beginning of T1 until the end of T2B Synchronous DRAM: From the beginning of Tr until completion of access 1: The intervals over which DACK1 is asserted on the relevant bus interfaces are as indicated below. Normal space: The same as for $\overline{RD}$ or $\overline{WRxx}$ MPX-I/O: The same as for $\overline{RD}$ or $\overline{WRxx}$ Only set this bit to 1 if the area of memory that is the target for transfer at the time of DACK1 assertion is in a normal space or the MPX-I/O space.

Bit	Bit Name	Initial Value	R/W	Description
0	DACK0TMG	0	R/W	DACK0 Pin Timing Select This bit controls timing of the assertion of the DACK0 pin. 0: The intervals over which DACK0 is asserted on the relevant bus interfaces are as indicated below. Normal space: From the beginning of T1 until the end of T2 MPX-I/O: From the beginning of T1 until the end of T2 SRAM with byte selection: From the beginning of Th until the end of Tf Burst ROM: From the beginning of T1 until the end of T2B Synchronous DRAM: From the beginning of Tr until completion of access 1: The intervals over which DACK0 is asserted on the relevant bus interfaces are as indicated below. Normal space: The same as for $\overline{RD}$ or $\overline{WRxx}$ MPX-I/O: The same as for $\overline{RD}$ or $\overline{WRxx}$ Only set this bit to 1 if the area of memory that is the target for transfer at the time of DACK0 assertion is in a normal space or the MPX-I/O space.

22.2 Pull-Up MOS Control by Pin Function

Table 22.9 shows the pull-up MOS control by pin function and the pull-up MOS control in each operating mode.

Table 22.9 Pull-Up MOS Control

Pin Function	Power-On Reset	Manual Reset	Software Standby	Sleep	When Oscillation Stop is Detected	When POE Function is Used	Normal Operation
I/O port input other than PB12 and PB13	Off	On/off	On/off	On/off	On/off	On/off	On/off
$\overline{\text{BREQ}}$ and $\overline{\text{WAIT}}$ input (BSC)							
DREQ0 to DREQ3 input (DMAC)							
IRQ0 to IRQ7 input (INTC)							
$\overline{\text{MRES}}$ input (System control)							
$\overline{\text{POE0}}$ to $\overline{\text{POE4}}$, and $\overline{\text{POE8}}$ input (POE2)							
RXD0 to RXD4 input (SCI, SCIF)							
SCK0 to SCK4 input (SCI, SCIF)							
CRx0 input (RCAN-ET)							
$\overline{\text{ADTRG}}$ input (ADC)							
SSLO and RSPCR input (RSPI)							
MISO and MOSI input (RSPI)							
LNKSTA and COL input (Ether)							
CRS and RX_CLK input (Ether)							
MII_RXD0 to MII_RXD3 input (Ether)							
RX_ER and RX_DV input (Ether)							
TX_CLK and MDIO input (Ether)							

Pin Function	Power-On Reset	Manual Reset	Software Standby	Sleep	When Oscillation Stop is Detected	When POE Function is Used	Normal Operation
I/O port output	Off	On/off*	On/off*	On/off*	On/off*	On/off*	On/off*
Address output, CK output, $\overline{RD}$ output (BSC)							
$\overline{WRHH}$ and $\overline{WRHL}$ output (BSC)							
$\overline{WRH}$ and $\overline{WRL}$ output (BSC)							
DQMUU and DQMUL output (BSC)							
DQMLU and DQMLL output (BSC)							
RD/$\overline{WR}$, and $\overline{CS0}$ to $\overline{CS7}$ output (BSC)							
$\overline{BS}$, $\overline{FRAME}$, and $\overline{AH}$ output (BSC)							
$\overline{BACK}$ and $\overline{REFOUT}$ output (BSC)							
CKE, $\overline{CASU}$, and $\overline{CASL}$ output (BSC)							
$\overline{RASU}$ and $\overline{RASL}$ output (BSC)							
DACK0 to DACK3 output (DMAC)							
TEND0 to TEND3 output (DMAC)							
$\overline{IRQOUT}$ output (INTC)							
$\overline{UBCTRG}$ output (UBC)							
TXD0 to TXD4 output (SCI, SCIF)							
SCK0 to SCK4 output (SCI, SCIF)							
CTx0 output (RCAN-ET)							
SSL0 to SSL3 and RSPCK output (RSPI)							
MISO and MOSI output (RSPI)							
$\overline{AUDSYNC}$ and AUDCK output (AUD)							
AUDATA0 to AUDATA3 output (AUD)							
WOL and EXOUT output (Ether)							

Pin Function	Power-On Reset	Manual Reset	Software Standby	Sleep	When Oscillation Stop is Detected	When POE Function is Used	Normal Operation
PB2 and PB3 input	Off	Off	Off	Off	Off	Off	Off
Data bus input/output (BSC)							
TIOC3AS and TIOC3BS input/output (MTU2S)							
TIOC3CS and TIOC3DS input/output (MTU2S)							
TIOC4AS and TIOC4BS input/output (MTU2S)							
TIOC4CS and TIOC4DS input/output (MTU2S)							
TIC5US, TIC5VS, and TIC5WS input (MTU2S)							
TCLKA and TCLKB input (MTU2)							
TCLKC and TCLKD input (MTU2)							
TIOC0A and TIOC0B input/output (MTU2)							
TIOC0C and TIOC0D input/output (MTU2)							
TIOC1A and TIOC1B input/output (MTU2)							
TIOC2A and TIOC2B input/output (MTU2)							
TIOC3C and TIOC3D input/output (MTU2)							
TIOC4A and TIOC4B input/output (MTU2)							
TIOC4C and TIOC4D input/output (MTU2)							
TIC5U, TIC5V, and TIC5W input (MTU2)							
SCL and SDA input/output (IIC)							
MDC and TX_EN output (Ether)							
MII_TXD0 to MII_TXD3 output (Ether)							
TX_ER and MDIO output (Ether)							

[Legend]

Off: Input pull-up MOS is always off.

On/off: Input pull-up MOS is on when the value of pull-up MOS control register is 1 and the pin is in input state or high impedance and off in other states.

On/off*: Input pull-up MOS is on when the value of pull-up MOS control register is 1 and the pin is in high impedance and off in other states.

Note: * For SCK (SCI, SCIF), MDIO (Ether), and MOSI, MISO, RSPCK, and SSL0 (RSPI) functions, when the pull-up MOS control register value is 1, if the input/output is switched, the on/off of the pull-up MOS also switched.

22.3 Usage Notes

1. In this LSI, the same function is available as a multiplexed function on multiple pins. This approach is intended to increase the number of selectable pin functions and to allow the easier design of boards. Note the following points when two or more pins are specified for one function.

- When the pin function is input

  Signals input to several pins are formed as one signal through OR or AND logic and the signal is transmitted into the LSI. Therefore, a signal that differs from the input signals may be transmitted to the LSI depending on the input signals in other pins that have the same functions. Table 22.10 shows the transmit forms of input functions allocated to several pins. When using one of the functions shown below in multiple pins, use it with care of signal polarity considering the transmit forms.

Table 22.10 Transmission Format of Input Function Allocated on Multiple Pins

OR Type	AND Type
TCLKA, TCLKB, TCLKC, TCLKD (MTU2)	IRQ0 to IRQ7 (INTC)
TIOC0A, TIOC0B, TIOC0C, TIOC0D (MTU2)	DREQ0, DREQ1 (DMAC)
TIOC1A, TIOC1B, TIOC2A (MTU2)	$\overline{\text{ADTRG}}$ (ADC)
TIC5U, TIC5V, TIC5W (MTU2)	$\overline{\text{WAIT}}$, $\overline{\text{BREQ}}$ (BSC)
TIOC3AS, TIOC3BS, TIOC3CS, TIOC3DS (MTU2S)	CRx0 (RCAN-ET)
TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS (MTU2S)	
SCK0 to SCK3, RXD0 to RXD3 (SCI, SCIF)	
$\overline{\text{POE0}}$, $\overline{\text{POE4}}$, $\overline{\text{POE8}}$ (POE2)	
SSLO, MISO, MOSI, RSPCK (RSPI)	
LNKSTA, COL, CRS, MDIO, RX_CLK (Ether)	
MII_RXD0 to MII_RXD3, RX_ER, RX_DV, TX_CLK (Ether)	

OR Type: Signals input to several pins are formed as one signal through OR logic and the signal is transmitted into the LSI.

AND Type: Signals input to several pins are formed as one signal through AND logic and the signal is transmitted into the LSI.

- When the pin function is output

  Each selected pin can output the same function.

2. When the port input is switched from the low level to the DREQ edge or the IRQ edge for the pins that are multiplexed with I/O and DREQ or IRQ, the corresponding edge is detected.
3. Do not set functions other than settable functions. Otherwise, correct operation cannot be guaranteed.

Section 23 I/O Ports

This LSI has six ports: A, B, C, D, E, and F. Port A is a 22-bit, port C is a 16-bit, port D is a 32-bit, and port E is a 16-bit I/O ports.

Port B has a 14-bit I/O port and a 2-bit input-only port. Port F is an 8-bit input-only port.

All port pins are multiplexed with other pin functions. The functions of the multiplex pins are selected by means of the pin function controller (PFC).

Each port is provided with data registers for storing the pin data.

23.1 Port A

Port A is an I/O port with 22 pins shown in figure 23.1.

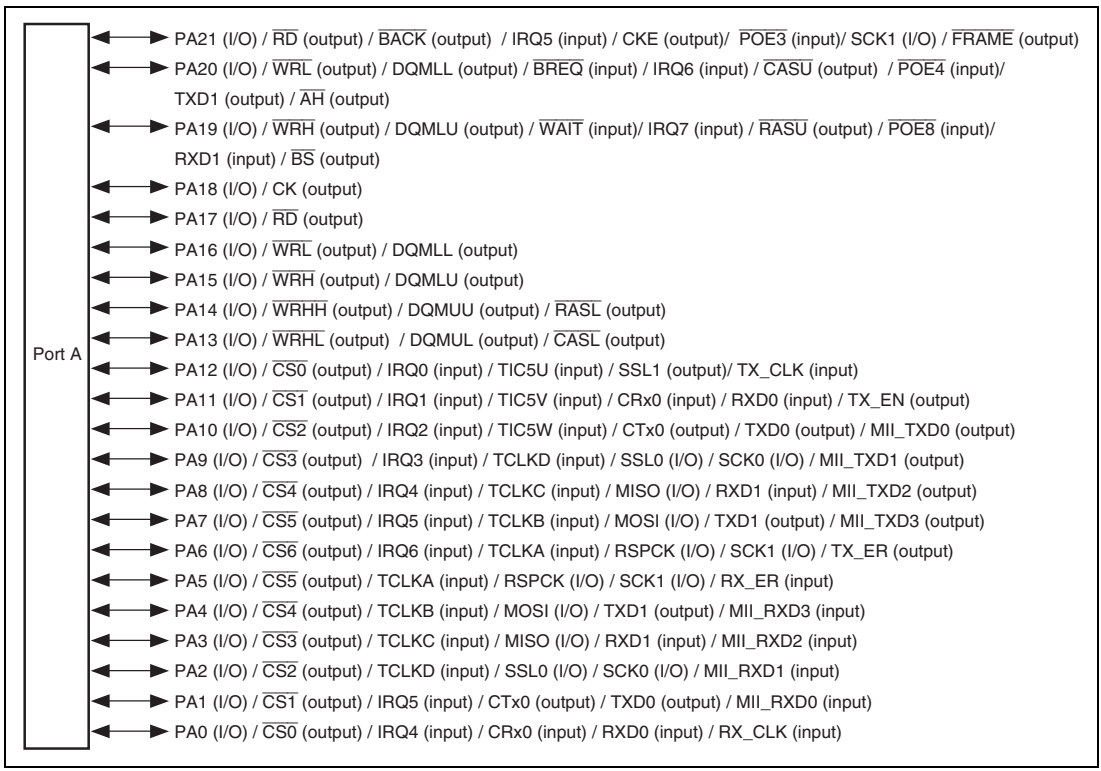


Figure 23.1 Port A

23.1.1 Register Descriptions

Port A has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

Table 23.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register H	PADRH	R/W	H'0000	H'FFFE3800	8, 16, 32
Port A data register L	PADRL	R/W	H'0000	H'FFFE3802	8, 16
Port A port register H	PAPRH	R	—	H'FFFE381C	8, 16, 32
Port A port register L	PAPRL	R	—	H'FFFE381E	8, 16

23.1.2 Port A Data Registers H and L (PADRH and PADRL)

PADRH and PADRL are 16-bit readable/writable registers that store port A data. Bits PA21DR to PA0DR correspond to pins PA21 to PA0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PADRH or PADRL, the value is output directly from the pin, and if PADRH or PADRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PADRH or PADRL is read, the pin state, not the register value, is returned directly. If a value is written to PADRH or PADRL, although that value is written into PADRH or PADRL, it does not affect the pin state.

Table 23.2 summarizes read/write operations of port A data register.

- Port A data register H (PADRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA21 DR	PA20 DR	PA19 DR	PA18 DR	PA17 DR	PA16 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	PA21DR	0	R/W	See table 23.2.
4	PA20DR	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
3	PA19DR	0	R/W	See table 23.2.
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

- Port A data register L (PADRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 23.2.
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

Table 23.2 Port A Data Registers H and L (PADRH and PADRL) Read/Write Operations

PAIORH, PAIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PADRH and PADRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PADRH and PADRL, but it has no effect on pin state.
1	General output	PADRH or PADRL value	The value written is output from the pin.
	Other than general output	PADRH or PADRL value	Can write to PADRH and PADRL, but it has no effect on pin state.

23.1.3 Port A Port Registers H and L (PAPRH and PAPRL)

PAPRH and PAPRL are 16-bit read-only registers, which return the states of the pins. However, when the RSPI function is selected for PA12 and the Ethernet functions are selected for PA11 to PA6, the states of the corresponding pins cannot be read out. In this LSI, bits PA21PR to PA0PR correspond to pins PA21 to PA0, respectively (description of multiplexed functions are abbreviated here).

- Port A port register H (PAPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA21PR	PA20PR	PA19PR	PA18PR	PA17PR	PA16PR
Initial value:	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
5	PA21PR	Pin state	R	The pin state is returned. These bits cannot be modified.
4	PA20PR	Pin state	R	
3	PA19PR	Pin state	R	
2	PA18PR	Pin state	R	
1	PA17PR	Pin state	R	
0	PA16PR	Pin state	R	

- Port A port register L (PAPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PR	PA14 PR	PA13 PR	PA12 PR	PA11 PR	PA10 PR	PA9 PR	PA8 PR	PA7 PR	PA6 PR	PA5 PR	PA4 PR	PA3 PR	PA2 PR	PA1 PR	PA0 PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PA15PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PA14PR	Pin state	R	
13	PA13PR	Pin state	R	
12	PA12PR	Pin state	R	
11	PA11PR	Pin state	R	
10	PA10PR	Pin state	R	
9	PA9PR	Pin state	R	
8	PA8PR	Pin state	R	
7	PA7PR	Pin state	R	
6	PA6PR	Pin state	R	
5	PA5PR	Pin state	R	
4	PA4PR	Pin state	R	
3	PA3PR	Pin state	R	
2	PA2PR	Pin state	R	
1	PA1PR	Pin state	R	
0	PA0PR	Pin state	R	

23.2 Port B

Port B is an I/O port with 16 pins shown in figure 23.2.

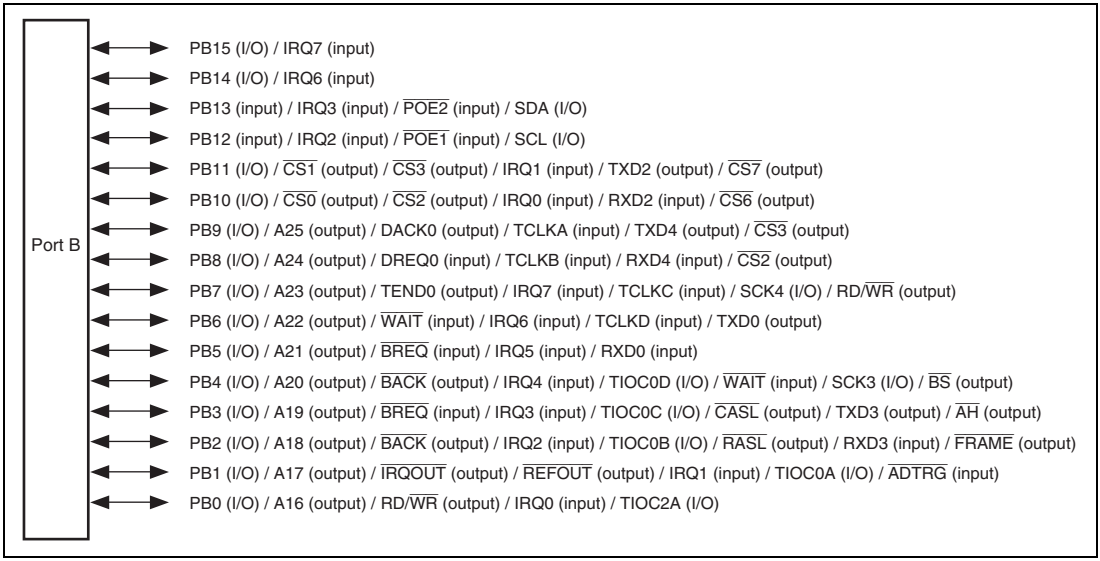


Figure 23.2 Port B

23.2.1 Register Descriptions

Port B has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

Table 23.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register L	PBDRL	R/W	H'0000	H'FFFE3882	8, 16
Port B port register L	PBPRL	R	—	H'FFFE389E	8, 16

23.2.2 Port B Data Register L PBDRL)

PBDRL is a 16-bit readable/writable register that stores port B data. Bits PB15DR, PB0DR correspond to pins PB15 to PB0, respectively (description of multiplexed functions are abbreviated here. When a pin function is general output, if a value is written to PBDRL, the value is output directly from the pin, and if PBDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PBDRL is read, the pin state, not the register value, is returned directly. If a value is written to PBDRL, although that value is written into PBDRL, it does not affect the pin state. Note that pins PB13 and PB12 do not function as general output pins, and only functions as general input pins.

Table 23.4 summarizes read/write operations of port B data register.

- Port B data register L (PBDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15 DR	PB14 DR	PB13 DR	PB12 DR	PB11 DR	PB10 DR	PB9 DR	PB8 DR	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	PB0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PB15DR	0	R/W	See table 23.4
14	PB14DR	0	R/W	
13	PB13DR	0	R/W	
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 23.4 Port B Data Register L (PBDRL) Read/Write Operations

PBIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PBDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PBDRL, but it has no effect on pin state.
1	General output	PBDRL value	The value written is output from the pin.
	Other than general output	PBDRL value	Can write to PBDRL, but it has no effect on pin state.

23.2.3 Port B Port Register L (PBPRL)

PBPRL is a 16-bit read-only register, which returns the states of the pins. However, when the SCIF function is selected for PB3, and the TE bit in SCSCR and the SPB2IO bit in SCSPTR are 0, the states of the corresponding pins cannot be read out. In this LSI, bits PB15PR to PB0PR correspond to pins PB15 to PB0, respectively (description of multiplexed functions are abbreviated here).

- Port B port register L (PBPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PB15PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PB14PR	Pin state	R	
13	PB13PR	Pin state	R	
12	PB12PR	Pin state	R	
11	PB11PR	Pin state	R	
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	
6	PB6PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
5	PB5PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
4	PB4PR	Pin state	R	
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	PB0PR	Pin state	R	

23.3 Port C

Port C is an I/O port with 16 pins shown in figure 23.3.

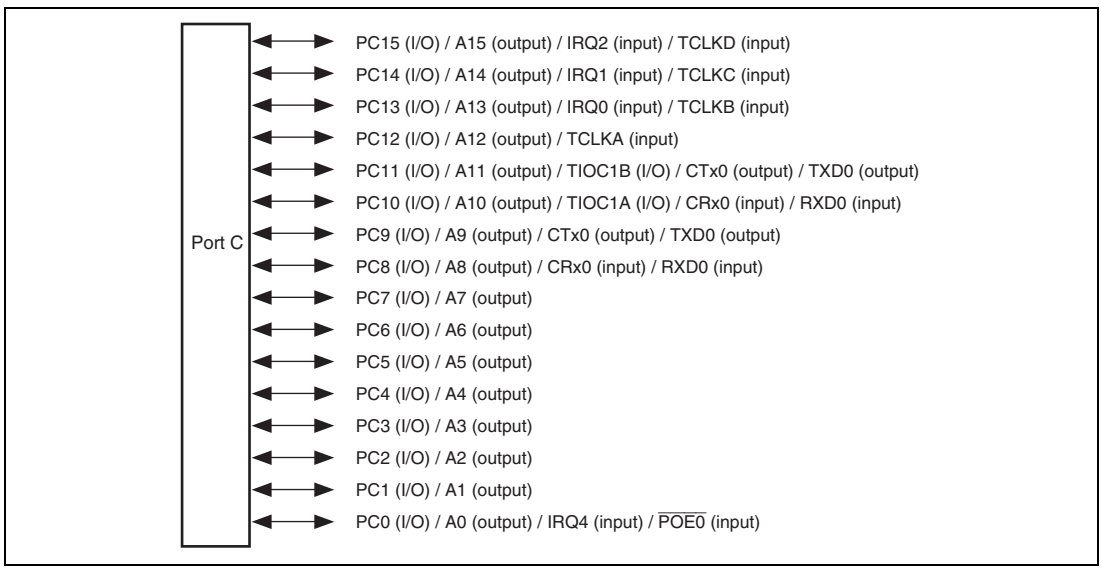


Figure 23.3 Port C

23.3.1 Register Descriptions

Port C has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

Table 23.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register L	PCDRL	R/W	H'0000	H'FFFE3902	8, 16
Port C port register L	PCPRL	R	—	H'FFFE391E	8, 16

23.3.2 Port C Data Register L (PCDRL)

PCDRL is a 16-bit readable/writable register that store port C data. Bits PC15DR to PC0DR correspond to pins PC15 to PC0 (description of multiplexed functions are abbreviated) respectively. When a pin function is general output, if a value is written to PCDRL, the value is output directly from the pin, and if PCDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PCDRL is read, the pin state, not the register value, is returned directly. If a value is written to PCDRL, although that value is written into PCDRL, it does not affect the pin state.

Table 23.6 summarizes read/write operations of port C data register.

- Port C data register L (PCDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PC15DR	0	R/W	See table 23.6.
14	PC14DR	0	R/W	
13	PC13DR	0	R/W	
12	PC12DR	0	R/W	
11	PC11DR	0	R/W	
10	PC10DR	0	R/W	
9	PC9DR	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
8	PC8DR	0	R/W	See table 23.6.
7	PC7DR	0	R/W	
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

Table 23.6 Port C Data Register L (PCDRL) Read/Write Operations

PCIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PCDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PCDRL, but it has no effect on pin state.
1	General output	PCDRL value	The value written is output from the pin.
	Other than general output	PCDRL value	Can write to PCDRL, but it has no effect on pin state.

23.3.3 Port C Port Register L (PCPRL)

PCPRL is a 16-bit read-only register, which always returns the states of the pins regardless of the PFC setting. In this LSI, bits PC15PR to PC0PR correspond to pins PC15 to PC0, respectively (description of multiplexed functions are abbreviated here).

- Port C port register L (PCPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PC15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PC14PR	Pin state	R	
13	PC13PR	Pin state	R	
12	PC12PR	Pin state	R	
11	PC11PR	Pin state	R	
10	PC10PR	Pin state	R	
9	PC9PR	Pin state	R	
8	PC8PR	Pin state	R	
7	PC7PR	Pin state	R	
6	PC6PR	Pin state	R	
5	PC5PR	Pin state	R	
4	PC4PR	Pin state	R	
3	PC3PR	Pin state	R	
2	PC2PR	Pin state	R	
1	PC1PR	Pin state	R	
0	PC0PR	Pin state	R	

23.4 Port D

Port D is an I/O port with 32 pins shown in figure 23.4.

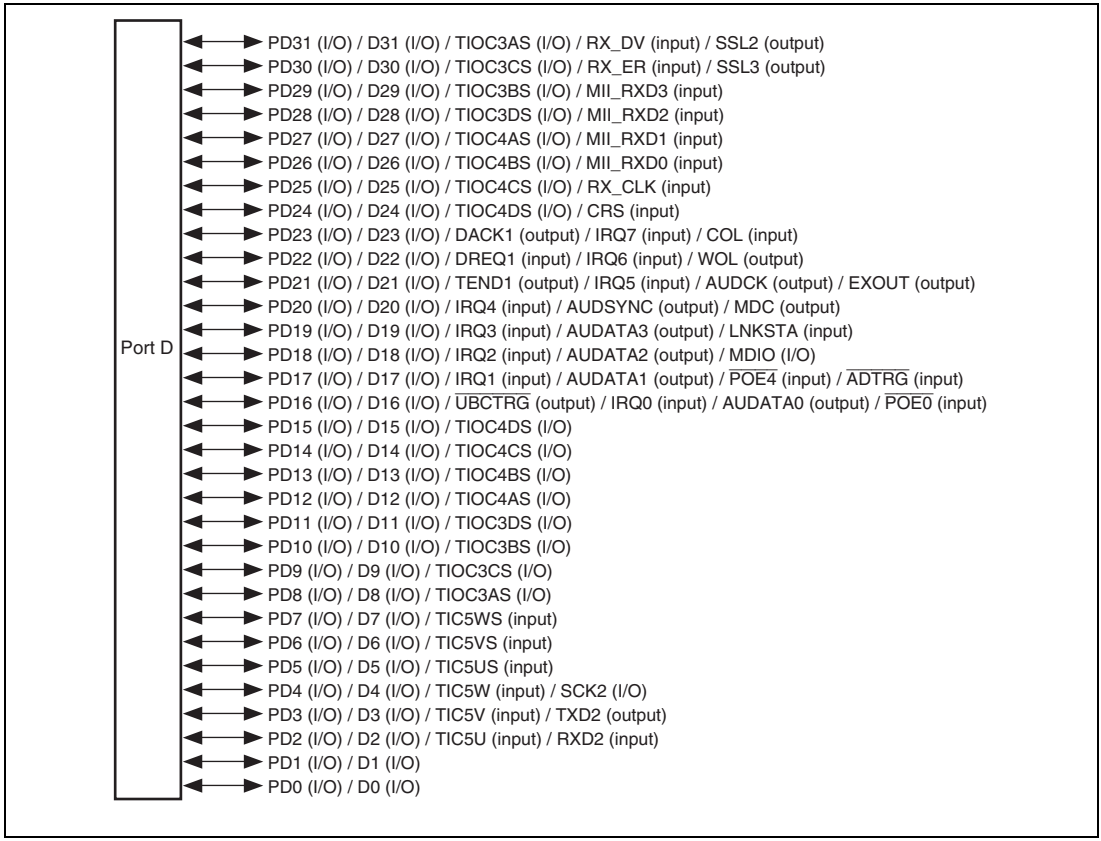


Figure 23.4 Port D

23.4.1 Register Descriptions

Port D has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

Table 23.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register H	PDDRH	R/W	H'0000	H'FFFE3980	8, 16, 32
Port D data register L	PDDRL	R/W	H'0000	H'FFFE3982	8, 16
Port D port register H	PDPRH	R	—	H'FFFE399C	8, 16, 32
Port D port register L	PDPRL	R	—	H'FFFE399E	8, 16

23.4.2 Port D Data Registers H and L (PDDRH and PDDRL)

PDDRH and PDDRL are 16-bit readable/writable registers that store port D data. In this LSI, bits PD31DR, to PD0DR correspond to pins PD31 to PD0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PDDRH or PDDRL, the value is output directly from the pin, and if PDDRH or PDDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PDDRH or PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRH or PDDRL, although that value is written into PDDRH or PDDRL, it does not affect the pin state.

Table 23.8 summarizes read/write operations of port D data register.

- Port D data register H (PDDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 DR	PD30 DR	PD29 DR	PD28 DR	PD27 DR	PD26 DR	PD25 DR	PD24 DR	PD23 DR	PD22 DR	PD21 DR	PD20 DR	PD19 DR	PD18 DR	PD17 DR	PD16 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD31DR	0	R/W	See table 23.8.
14	PD30DR	0	R/W	
13	PD29DR	0	R/W	
12	PD28DR	0	R/W	
11	PD27DR	0	R/W	
10	PD26DR	0	R/W	
9	PD25DR	0	R/W	
8	PD24DR	0	R/W	
7	PD23DR	0	R/W	
6	PD22DR	0	R/W	
5	PD21DR	0	R/W	
4	PD20DR	0	R/W	
3	PD19DR	0	R/W	
2	PD18DR	0	R/W	
1	PD17DR	0	R/W	
0	PD16DR	0	R/W	

- Port D data register L (PDDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 DR	PD14 DR	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9 DR	PD8 DR	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	PD1 DR	PD0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PD15DR	0	R/W	See table 23.8.
14	PD14DR	0	R/W	
13	PD13DR	0	R/W	
12	PD12DR	0	R/W	
11	PD11DR	0	R/W	
10	PD10DR	0	R/W	
9	PD9DR	0	R/W	
8	PD8DR	0	R/W	
7	PD7DR	0	R/W	
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

Table 23.8 Port D Data Registers H and L (PDDRH and PDDRL) Read/Write Operations

- PDDRL bits 15 to 0

PDIORH, PDIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PDDRH and PDDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PDDRH and PDDRL, but it has no effect on pin state.
1	General output	PDDRH or PDDRL value	The value written is output from the pin.
	Other than general output	PDDRH or PDDRL value	Can write to PDDRH and PDDRL, but it has no effect on pin state.

23.4.3 Port D Port Registers H and L (PDPRH and PDPRL)

PDPRH and PDPRL are 16-bit read-only registers, which return the states of the pins. However, when the RSPI functions are selected for PD31 and PD30 and the Ethernet functions are selected for PD22 to PD20, the states of the corresponding pins cannot be read out. In this LSI, bits PD31PR to PD0PR correspond to pins PD31 to PD0, respectively (description of multiplexed functions are abbreviated here).

- Port D port register H (PDPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31PR	PD30PR	PD29PR	PD28PR	PD27PR	PD26PR	PD25PR	PD24PR	PD23PR	PD22PR	PD21PR	PD20PR	PD19PR	PD18PR	PD17PR	PD16PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PD31PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PD30PR	Pin state	R	
13	PD29PR	Pin state	R	
12	PD28PR	Pin state	R	
11	PD27PR	Pin state	R	
10	PD26PR	Pin state	R	
9	PD25PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
8	PD24PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
7	PD23PR	Pin state	R	
6	PD22PR	Pin state	R	
5	PD21PR	Pin state	R	
4	PD20PR	Pin state	R	
3	PD19PR	Pin state	R	
2	PD18PR	Pin state	R	
1	PD17PR	Pin state	R	
0	PD16PR	Pin state	R	

- Port D port register L (PDPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PD15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PD14PR	Pin state	R	
13	PD13PR	Pin state	R	
12	PD12PR	Pin state	R	
11	PD11PR	Pin state	R	
10	PD10PR	Pin state	R	
9	PD9PR	Pin state	R	
8	PD8PR	Pin state	R	
7	PD7PR	Pin state	R	
6	PD6PR	Pin state	R	
5	PD5PR	Pin state	R	
4	PD4PR	Pin state	R	
3	PD3PR	Pin state	R	
2	PD2PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
1	PD1PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
0	PD0PR	Pin state	R	

23.5 Port E

Port E is an I/O port with 16 pins shown in figure 23.5

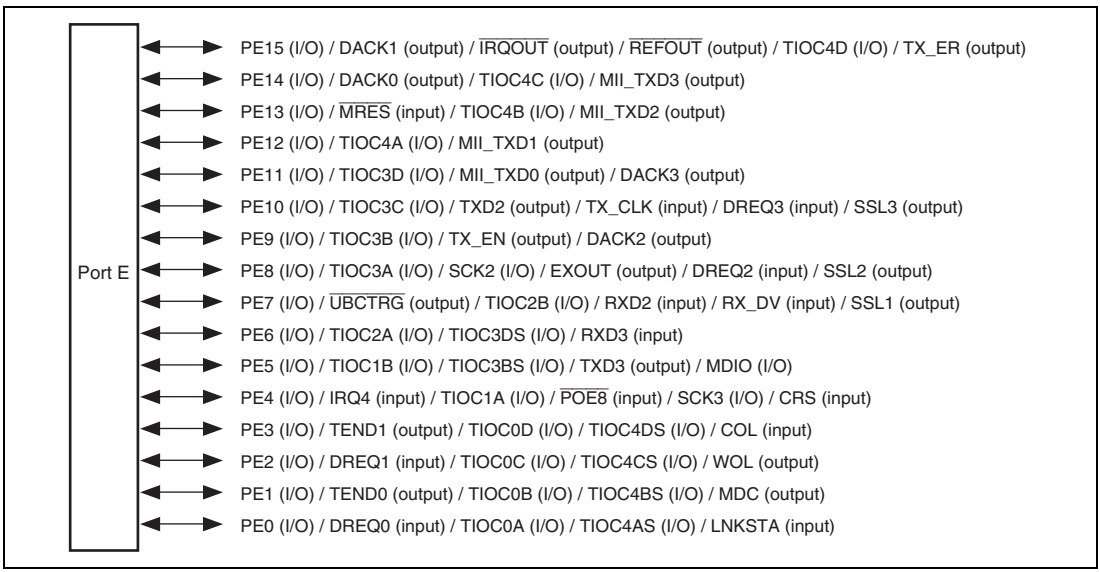


Figure 23.5 Port E

23.5.1 Register Descriptions

Port E has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

Table 23.9 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register L	PEDRL	R/W	H'0000	H'FFFE3A02	8, 16
Port E port register L	PEPRL	R	—	H'FFFE3A1E	8, 16

23.5.2 Port E Data Register L (PEDRL)

PEDRL is a 16-bit readable/writable register that stores port E data. In this LSI, bits PE15DR to PE0DR correspond to pins PE15 to PE0, respectively (description of multiplexed functions are abbreviated here). When a pin function is general output, if a value is written to PEDRL, the value is output directly from the pin, and if PEDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PEDRL is read, the pin state, not the register value, is returned directly. If a value is written to PEDRL, although that value is written into PEDRL, it does not affect the pin state.

Table 23.10 summarizes read/write operations of port E data register.

- Port E data register L (PEDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PE15DR	0	R/W	See table 23.10.
14	PE14DR	0	R/W	
13	PE13DR	0	R/W	
12	PE12DR	0	R/W	
11	PE11DR	0	R/W	
10	PE10DR	0	R/W	
9	PE9DR	0	R/W	
8	PE8DR	0	R/W	
7	PE7DR	0	R/W	
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

Table 23.10 Port E Data Register L (PEDRL) Read/Write Operations

PEIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PEDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PEDRL, but it has no effect on pin state.
1	General output	PEDRL value	The value written is output from the pin.
	Other than general output	PEDRL value	Can write to PEDRL, but it has no effect on pin state.

23.5.3 Port E Port Register L (PEPRL)

PEPRL is a 16-bit read-only register, which returns the states of the pins. However, when the TE bit in SCSCR and the SPB2IO bit in SCSPTR are 0, and the RSPI functions are selected for PE10, PE8, and PE7, the Ethernet functions are selected for PE15 to PE11, PE9, PE8, PE2, and PE1, and the SCIF function is selected for PE5, the states of the corresponding pins cannot be read out. In this LSI, bits PE15PR to PE0PR correspond to pins PE15 to PE0, respectively (description of multiplexed functions are abbreviated here).

- Port E port register L (PEPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
Initial value:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PE15PR	Pin state	R	The pin state is returned. These bits cannot be modified.
14	PE14PR	Pin state	R	
13	PE13PR	Pin state	R	
12	PE12PR	Pin state	R	
11	PE11PR	Pin state	R	
10	PE10PR	Pin state	R	
9	PE9PR	Pin state	R	
8	PE8PR	Pin state	R	
7	PE7PR	Pin state	R	
6	PE6PR	Pin state	R	

Bit	Bit Name	Initial Value	R/W	Description
5	PE5PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
4	PE4PR	Pin state	R	
3	PE3PR	Pin state	R	
2	PE2PR	Pin state	R	
1	PE1PR	Pin state	R	
0	PE0PR	Pin state	R	

23.6 Port F

Port F is an I/O port with 8 pins shown in figure 23.6.

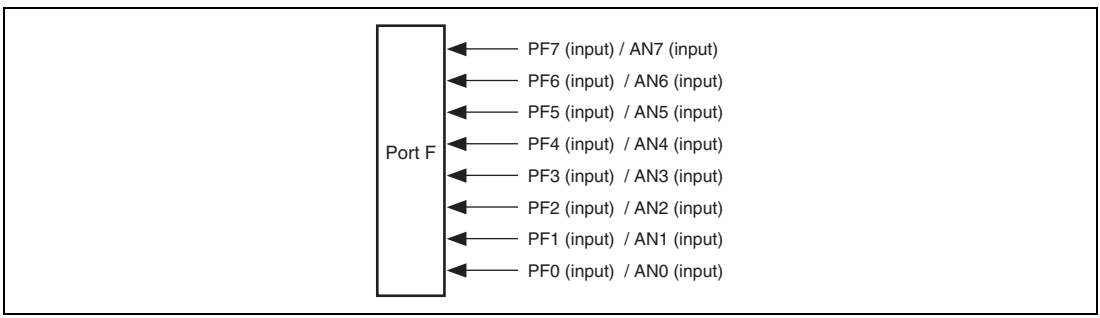


Figure 23.6 Port F

23.6.1 Register Descriptions

Port F has the following registers. See section 32, List of Registers for details on the register address and states in each operating mode.

Table 23.11 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register L	PFDRL	R	—	H'FFFE3A82	8, 16

23.6.2 Port F Data Register L (PFDRL)

PFDRL is a 16-bit read-only register that stores port F data. In this LSI, bits PF7DR to PF0DR correspond to pins PF7 to PF0, respectively (description of multiplexed functions are abbreviated here).

Even if a value is written to PFDR, the value is not written into PFDR, and it does not affect the pin state. If PFDR is read, the pin state, not the register value, is returned directly. However, when sampling the analog input of A/D converter, 1 is read. Table 23.12 summarizes read/write operations of port F data register.

- Port F data register L (PFDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PF7DR	Pin state	R	See table 23.12.
6	PF6DR	Pin state	R	
5	PF5DR	Pin state	R	
4	PF4DR	Pin state	R	
3	PF3DR	Pin state	R	
2	PF2DR	Pin state	R	
1	PF1DR	Pin state	R	
0	PF0DR	Pin state	R	

Table 23.12 Port F Data Register L (PFDRL) Read/Write Operations

Pin Function	Read	Write
General input	Pin state	Ignored (no effect on pin state)
ANn input	1	Ignored (no effect on pin state)

23.7 Usage Notes

23.7.1 Handling of Unused pins

Levels on unused pins of port F should be fixed by connection to AV_{CC} or AV_{SS} via resistances. For handling of the NMI, USD+, USD-, EXTAL, XTAL, USBEXTAL, USBXTAL, $\overline{\text{WDTOVF}}$, $\overline{\text{TRST}}$, TMS, TCK, TDO, and TDI pins, follow the instructions in the sections on the corresponding modules. Other unused pins should be connected to $V_{CC}Q$ or GND via resistors to fix high or low levels on the pins.

Section 24 USB Function Module (USB)

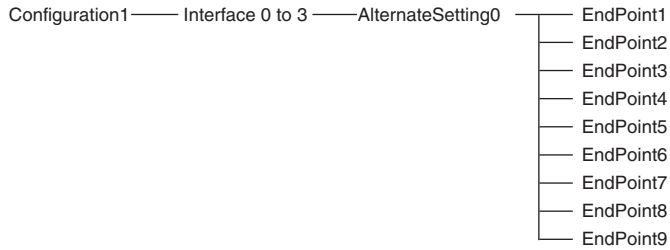
This LSI incorporates a USB function module (USB).

24.1 Features

- Automatic processing of USB protocol with on-chip protocol processor and transceiver conforming to USB2.0

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Full-speed (12 Mbps)
- Endpoint configuration

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA/DTC Transfer
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control IN	16	16	—
	EP0o	Control OUT	16	16	—
Endpoint 1	EP1	Bulk OUT	64	128	Possible
Endpoint 2	EP2	Bulk IN	64	128	Possible
Endpoint 3	EP3	Interrupt IN	16	16	—
Endpoint 4	EP4	Bulk OUT	64	128	Possible
Endpoint 5	EP5	Bulk IN	64	128	Possible
Endpoint 6	EP6	Interrupt IN	16	16	—
Endpoint 7	EP7	Bulk OUT	64	64	—
Endpoint 8	EP8	Bulk IN	64	64	—
Endpoint 9	EP9	Interrupt IN	16	16	—



- Interrupt requests: Generates various interrupt signals necessary for USB transmission/reception
- Clock*: External input (48 MHz)

  Internal input (only when 12-MHz EXTAL is used)
- Power-down mode

  Power consumption can be reduced by stopping the protocol-processor internal clock when USB cable is disconnected.
- Power mode: Self-powered mode

Note: * Use the USBSEL bit in the standby control register 6 (STBCR6) for selection of the clock. For details, see section 30.3.6, Standby Control Register 6 (STBCR6).

Figure 24.1 shows a block diagram of the USB.

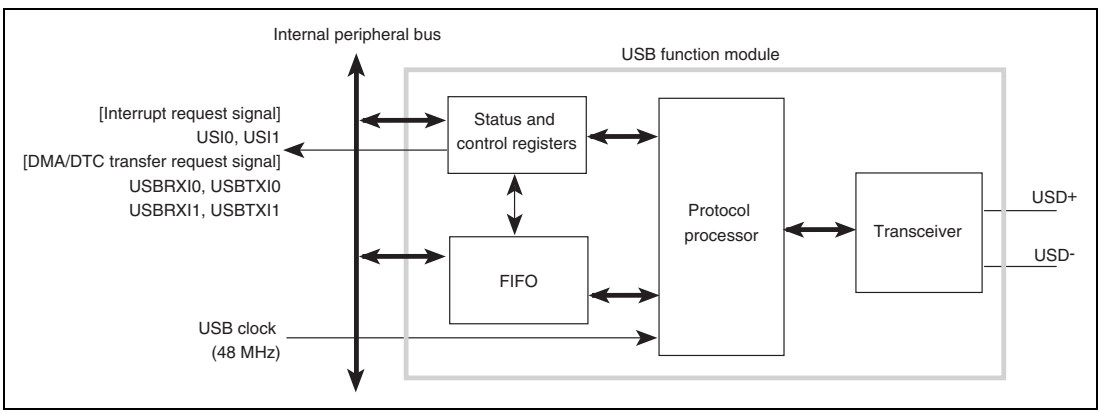


Figure 24.1 Block Diagram of USB

24.2 Pin Configuration

Table 24.1 lists input/output pins and their functions of the USB.

Table 24.1 Pin Configuration

Pin Name	I/O	Function
VBUS	Input	USB cable connection monitor pin
USD+	I/O	USB data input/output pin
USD–	I/O	USB data input/output pin
DrVcc	Input	USB on-chip transceiver power supply pin (3.0 to 3.6V, DrVcc = VccQ)
DrVss	Input	USB on-chip transceiver ground pin (Connect to Vss)
USBEXTAL	Input	Connected to a 48-MHz resonator for USB
USBXTAL	Output	Connected to a 48-MHz resonator for USB
PUPD (PB15)	Output	Pull-up control

24.3 Register Descriptions

The USB has the following registers.

Table 24.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
USB interrupt flag register 0	USBIFR0	R/W	H'0x	H'FFFE7000	8
USB interrupt flag register 1	USBIFR1	R/W	H'00	H'FFFE7001	8
USB interrupt flag register 2	USBIFR2	R/W	H'06	H'FFFE7002	8
USB interrupt flag register 3	USBIFR3	R/W	H'06	H'FFFE7003	8
USB interrupt flag register 4	USBIFR4	R/W	H'04	H'FFFE7004	8
USB interrupt enable register 0	USBIER0	R/W	H'00	H'FFFE7008	8
USB interrupt enable register 1	USBIER1	R/W	H'00	H'FFFE7009	8
USB interrupt enable register 2	USBIER2	R/W	H'00	H'FFFE700A	8
USB interrupt enable register 3	USBIER3	R/W	H'00	H'FFFE700B	8
USB interrupt enable register 4	USBIER4	R/W	H'00	H'FFFE700C	8
USB interrupt select register 0	USBISR0	R/W	H'00	H'FFFE7010	8
USB interrupt select register 1	USBISR1	R/W	H'00	H'FFFE7011	8
USB interrupt select register 2	USBISR2	R/W	H'00	H'FFFE7012	8
USB interrupt select register 3	USBISR3	R/W	H'00	H'FFFE7013	8
USB interrupt select register 4	USBISR4	R/W	H'00	H'FFFE7014	8
USBEP0i data register	USBEPDR0i	W	Undefined	H'FFFE7020	8, 16, 32
USBEP0o data register	USBEPDR0o	R	Undefined	H'FFFE7024	8, 16, 32
USBEP0s data register	USBEPDR0s	R	Undefined	H'FFFE7028	8, 16, 32
USBEP1 data register	USBEPDR1	R	Undefined	H'FFFE7030	8, 16, 32
USBEP2 data register	USBEPDR2	W	Undefined	H'FFFE7034	8, 16, 32
USBEP3 data register	USBEPDR3	W	Undefined	H'FFFE7038	8, 16, 32
USBEP4 data register	USBEPDR4	R	Undefined	H'FFFE7040	8, 16, 32
USBEP5 data register	USBEPDR5	W	Undefined	H'FFFE7044	8, 16, 32
USBEP6 data register	USBEPDR6	W	Undefined	H'FFFE7048	8, 16, 32
USBEP7 data register	USBEPDR7	R	Undefined	H'FFFE7050	8, 16, 32
USBEP8 data register	USBEPDR8	W	Undefined	H'FFFE7054	8, 16, 32
USBEP9 data register	USBEPDR9	W	Undefined	H'FFFE7058	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
USBEP0o receive data size register	USBEPSZ0o	R	H'00	H'FFFE7080	8
USBEP1 receive data size register	USBEPSZ1	R	H'00	H'FFFE7081	8
USBEP4 receive data size register	USBEPSZ4	R	H'00	H'FFFE7082	8
USBEP7 receive data size register	USBEPSZ7	R	H'00	H'FFFE7083	8
USB data status register 0	USBDASTS0	R	H'00	H'FFFE7088	8
USB data status register 1	USBDASTS1	R	H'00	H'FFFE7089	8
USB data status register 2	USBDASTS2	R	H'00	H'FFFE708A	8
USB data status register 3	USBDASTS3	R	H'00	H'FFFE708B	8
USB trigger register 0	USBTRG0	W	H'00	H'FFFE7090	8
USB trigger register 1	USBTRG1	W	H'00	H'FFFE7091	8
USB trigger register 2	USBTRG2	W	H'00	H'FFFE7092	8
USB trigger register 3	USBTRG3	W	H'00	H'FFFE7093	8
USB FIFO clear register 0	USBFCLR0	W	H'00	H'FFFE7098	8
USB FIFO clear register 1	USBFCLR1	W	H'00	H'FFFE7099	8
USB FIFO clear register 2	USBFCLR2	W	H'00	H'FFFE709A	8
USB FIFO clear register 3	USBFCLR3	W	H'00	H'FFFE709B	8
USB endpoint stall register 0	USBEPSTL0	R/W	H'00	H'FFFE70A0	8
USB endpoint stall register 1	USBEPSTL1	R/W	H'00	H'FFFE70A1	8
USB endpoint stall register 2	USBEPSTL2	R/W	H'00	H'FFFE70A2	8
USB endpoint stall register 3	USBEPSTL3	R/W	H'00	H'FFFE70A3	8
USB stall status register 1	USBSTLSR1	R/W	H'00	H'FFFE70A9	8
USB stall status register 2	USBSTLSR2	R/W	H'00	H'FFFE70AA	8
USB stall status register 3	USBSTLSR3	R/W	H'00	H'FFFE70AB	8
USB DMA transfer setting register	USBDMAR	R/W	H'00	H'FFFE70B0	8
USB configuration value register	USBCVR	R	H'00	H'FFFE70B4	8
USB control register	USBCTLR	R/W	H'01	H'FFFE70B8	8
USB endpoint information register	USBEPIR	W	Undefined	H'FFFE70C0	8
USB transceiver test register 0	USBTRNTREG0	R/W	H'00	H'FFFE70D0	8
USB transceiver test register 1	USBTRNTREG1	R	H'00	H'FFFE70D1	8

24.3.1 USB Interrupt Flag Register 0 (USBIFR0)

Together with USB interrupt flag registers 1 to 4 (USBIFR1 to USBIFR4), USBIFR0 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with the USB interrupt enable register 0 (USBIER0). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, VBUSMN is a status bit, and cannot be cleared.

Bit:	7	6	5	4	3	2	1	0
	BRST	CFDN	-	-	SETC	SETI	VBUSMN	VBUSF
Initial value:	0	0	0	0	0	0	-	0
R/W:	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*	R	R/(W)*

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

Bit	Bit Name	Initial Value	R/W	Description
7	BRST	0	R/(W)*	Bus Reset This bit is set to 1 when the bus reset signal is detected on the USB bus.
6	CFDN	0	R/(W)*	Endpoint Information Loading Complete This bit is set to 1 when the data written to the USB endpoint information register (USBEPIR) has been set (loading completed) in this module. Upon completion of the endpoint information setting, this module can operate normally as USB.
5, 4	—	All 0	R	Reserved The write value should always be 0.
3	SETC	0	R/(W)*	Set_Configuration Command Detection This bit is set to 1 when the Set_Configuration command is detected.
2	SETI	0	R/(W)*	Set_Interface Command Detection This bit is set to 1 when the Set_Interface command is detected.

Bit	Bit Name	Initial Value	R/W	Description
1	VBUSMN	—	R	VBUS Pin Status Monitor VBUS pin status bit 0: VBUS pin = 0 1: VBUS pin = 1 VBUSMN is a status bit, and cannot be cleared. No VBUSMN interrupt request can be generated.
0	VBUSF	0	R/(W)*	USB Bus Connection/Disconnection Detection This bit is set to 1 when a function is connected to or disconnected from the USB bus. Use the VBUS pin of this module to detect connection/disconnection.

24.3.2 USB Interrupt Flag Register 1 (USBIFR1)

Together with USB interrupt flag registers 0, 2, 3, and 4 (USBIFR0, USBIFR2, USBIFR3, and USBIFR4), USBIFR1 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with the USB interrupt enable register 1 (USBIER1). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	SOF	SETUP TS	EP0oTS	EP0iTR	EP0iTS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SOF	0	R/(W)*	SOF Packet Detection This bit is set to 1 when the Start Of Frame (SOF) packet is detected.

Bit	Bit Name	Initial Value	R/W	Description
3	SETUPTS	0	R/(W)*	Setup Command Receive Complete This bit is set to 1 when endpoint 0 receives normally a setup command requiring decoding on the application side, and returns an ACK handshake to the host.
2	EP0oTS	0	R/(W)*	EP0o Receive Complete This bit is set to 1 when endpoint 0 receives data normally from the host, stores the data in the FIFO buffer, and returns an ACK handshake to the host.
1	EP0iTR	0	R/(W)*	EP0i Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 0 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
0	EP0iTS	0	R/(W)*	EP0i Transmit Complete This bit is set to 1 when data is transmitted to the host from endpoint 0 and an ACK handshake is returned.

24.3.3 USB Interrupt Flag Register 2 (USBIFR2)

Together with USB interrupt flag registers 0, 1, 3, and 4 (USBIFR0, USBIFR1, USBIFR3, and USBIFR4), USBIFR2 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with the USB interrupt enable register 2 (USBIER2). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, EP1FULL, EP2ALLEMP, and EP2EMPTY are status bits, and cannot be cleared.

Bit:	7	6	5	4	3	2	1	0
	-	-	EP3 TR	EP3 TS	EP2 TR	EP2 EMPTY	EP2 ALLEMP	EP1 FULL
Initial value:	0	0	0	0	0	1	1	0
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP3TR	0	R/(W)*	EP3 Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 3 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
4	EP3TS	0	R/(W)**	EP3 Transmit Complete This bit is set to 1 when data is transmitted to the host from endpoint 3 and an ACK handshake is returned.
3	EP2TR	0	R/(W)*	EP2 Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 2 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
2	EP2EMPTY	1	R	EP2 FIFO Empty This bit is set to 1 when at least one of the dual endpoint 2 transmit FIFO buffers is ready for transmit data to be written. EP2EMPTY is a status bit, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
1	EP2ALLEMP	1	R	EP2 FIFO All Empty This bit is set to 1 when both of the dual endpoint 2 transmit FIFO buffers are empty. EP2ALLEMP is a status bit, and cannot be cleared.
0	EP1FULL	0	R	EP1 FIFO Full This bit is set to 1 when endpoint 1 receives one packet of data normally from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. EP1FULL is a status bit, and cannot be cleared.

24.3.4 USB Interrupt Flag Register 3 (USBIFR3)

Together with USB interrupt flag registers 0, 1, 2, and 4 (USBIFR0, USBIFR1, USBIFR2, and USBIFR4), USBIFR3 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with the USB interrupt enable register 3 (USBIER3). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, EP4FULL, EP5ALLEMP, and EP5EMPTY are status bits, and cannot be cleared.

Bit:	7	6	5	4	3	2	1	0
	-	-	EP6 TR	EP6 TS	EP5 TR	EP5 EMPTY	EP5 ALLEMP	EP4 FULL
Initial value:	0	0	0	0	0	1	1	0
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP6TR	0	R/(W)*	EP6 Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 6 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	EP6TS	0	R/(W)*	EP6 Transmit Complete This bit is set to 1 when data is transmitted to the host from endpoint 6 and an ACK handshake is returned.
3	EP5TR	0	R/(W)*	EP5 Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 5 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
2	EP5EMPTY	1	R	EP5 FIFO Empty This bit is set to 1 when at least one of the dual endpoint 5 transmit FIFO buffers is ready for transmit data to be written. EP5EMPTY is a status bit, and cannot be cleared.
1	EP5ALLEMP	1	R	EP5 FIFO All Empty This bit is set to 1 when both of the dual endpoint 5 transmit FIFO buffers are empty. EP5ALLEMP is a status bit, and cannot be cleared.
0	EP4FULL	0	R	EP4 FIFO Full This bit is set to 1 when endpoint 4 receives one packet of data normally from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. EP4FULL is a status bit, and cannot be cleared.

24.3.5 USB Interrupt Flag Register 4 (USBIFR4)

Together with USB interrupt flag registers 0 to 3 (USBIFR0 to USBIFR3), USBIFR4 indicates interrupt status information required by the application. When an interrupt occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with the USB interrupt enable register 4 (USBIER4). Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, EP7FULL and EP8EMPTY are status bits, and cannot be cleared.

Bit:	7	6	5	4	3	2	1	0
	-	-	EP9TR	EP9TS	EP8TR	EP8EMPTY	-	EP7FULL
Initial value:	0	0	0	0	0	1	0	0
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP9TR	0	R/(W)*	EP9 Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 9 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
4	EP9TS	0	R/(W)*	EP9 Transmit Complete This bit is set to 1 when data is transmitted to the host from endpoint 9 and an ACK handshake is returned.
3	EP8TR	0	R/(W)*	EP8 Transfer Request This bit is set to 1 if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 8 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
2	EP8EMPTY	1	R	EP8 FIFO Empty This bit is set to 1 when the endpoint 8 transmit FIFO buffer is ready for transmit data to be written. EP8EMPTY is a status bit, and cannot be cleared.

Bit	Bit Name	Initial Value	R/W	Description
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	EP7FULL	0	R	EP7 FIFO Full This bit is set to 1 when endpoint 7 receives one packet of data normally from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. EP7FULL is a status bit, and cannot be cleared.

24.3.6 USB Interrupt Enable Register 0 (USBIER0)

USBIER0 enables the interrupt requests indicated in the USB interrupt flag register 0 (USBIFR0). When an interrupt flag is set while the corresponding bit in USBIER0 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the content of the USB interrupt select register 0 (USBISR0).

Bit:	7	6	5	4	3	2	1	0
	BRSTE	CFDNE	-	-	SETCE	SETIE	-	VBUSFE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	BRSTE	0	R/W	Bus reset
6	CFDNE	0	R/W	Endpoint information loading complete
5, 4	—	All 0	R	Reserved The write value should always be 0.
3	SETCE	0	R/W	Set_Configuration command detection
2	SETIE	0	R/W	Set_Interface command detection
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	VBUSFE	0	R/W	USB bus connection/disconnection detection

24.3.7 USB Interrupt Enable Register 1 (USBIER1)

USBIER1 enables the interrupt requests indicated in the USB interrupt flag register 1 (USBIFR1). When an interrupt flag is set while the corresponding bit in USBIER1 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the content of the USB interrupt select register 1 (USBISR1).

Bit:	7	6	5	4	3	2	1	0
	-	-	-	SOFE	SETUP TSE	EP0o TSE	EP0i TRE	EP0i TSE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SOFE	0	R/W	SOF packet detection
3	SETUPTSE	0	R/W	Setup command receive complete
2	EP0oTSE	0	R/W	EP0o receive complete
1	EP0iTRE	0	R/W	EP0i transfer request
0	EP0iTSE	0	R/W	EP0i transmit complete

24.3.8 USB Interrupt Enable Register 2 (USBIER2)

USBIER2 enables the interrupt requests indicated in the USB interrupt flag register 2 (USBIFR2). When an interrupt flag is set while the corresponding bit in USBIER2 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the content of the USB interrupt select register 2 (USBISR2).

Bit:	7	6	5	4	3	2	1	0
	-	-	EP3TRE	EP3TSE	EP2TRE	EP2EMPTYE	EP2ALLEMPE	EP1FULLE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP3TRE	0	R/W	EP3 transfer request
4	EP3TSE	0	R/W	EP3 transmit complete
3	EP2TRE	0	R/W	EP2 transfer request
2	EP2EMPTYE	0	R/W	EP2 FIFO empty
1	EP2ALLEMPE	0	R/W	EP2 FIFO all empty
0	EP1FULLE	0	R/W	EP1 FIFO full

24.3.9 USB Interrupt Enable Register 3 (USBIER3)

USBIER3 enables the interrupt requests indicated in the USB interrupt flag register 3 (USBIFR3). When an interrupt flag is set while the corresponding bit in USBIER3 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the content of the USB interrupt select register 3 (USBISR3).

Bit:	7	6	5	4	3	2	1	0
	-	-	EP6TRE	EP6TSE	EP5TRE	EP5EMPTYE	EP5ALLEMPE	EP4FULLE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP6TRE	0	R/W	EP6 transfer request
4	EP6TSE	0	R/W	EP6 transmit complete
3	EP5TRE	0	R/W	EP5 transfer request
2	EP5EMPTYE	0	R/W	EP5 FIFO empty
1	EP5ALLEMPE	0	R/W	EP5 FIFO all empty
0	EP4FULLE	0	R/W	EP4 FIFO full

24.3.10 USB Interrupt Enable Register 4 (USBIER4)

USBIER4 enables the interrupt requests indicated in the USB interrupt flag register 4 (USBIFR4). When an interrupt flag is set while the corresponding bit in USBIER4 is set to 1, an interrupt request is sent to the CPU. The interrupt vector number is determined by the content of the USB interrupt select register 4 (USBISR4).

Bit:	7	6	5	4	3	2	1	0
	-	-	EP9 TRE	EP9 TSE	EP8 TRE	EP8 EMPTYE	-	EP7 FULLE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP9TRE	0	R/W	EP9 transfer request
4	EP9TSE	0	R/W	EP9 transmit complete
3	EP8TRE	0	R/W	EP8 transfer request
2	EP8EMPTYE	0	R/W	EP8 FIFO empty
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	EP7FULLE	0	R/W	EP7 FIFO full

24.3.11 USB Interrupt Select Register 0 (USBISR0)

USBISR0 selects the vector numbers of the interrupt requests indicated in the USB interrupt flag register 0 (USBIFR0). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR0 is cleared to 0, the interrupt will be USI0. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR0 is set to 1, the interrupt will be USI1. If interrupts occur simultaneously, USI0 has priority by default.

Bit:	7	6	5	4	3	2	1	0
	BRSTS	CFDNS	-	-	SETCS	SETIS	-	VBUSFS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	BRSTS	0	R/W	Bus reset
6	CFDNS	0	R/W	Endpoint information loading complete
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	—	0	R	Reserved The write value should always be 0.
3	SETCS	0	R/W	Set_Configuration command detection
2	SETIS	0	R/W	Set_Interface command detection
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	VBUSFS	0	R/W	USB bus connection/disconnection detection

24.3.12 USB Interrupt Select Register 1 (USBISR1)

USBISR1 selects the vector numbers of the interrupt requests indicated in the USB interrupt flag register 1 (USBIFR1). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR1 is cleared to 0, the interrupt will be USI0. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR1 is set to 1, the interrupt will be USI1. If interrupts occur simultaneously, USI0 has priority by default.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	SOFS	SETUP TSS	EP0o TSS	EP0i TRS	EP0i TSS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SOFS	0	R/W	SOF packet detection
3	SETUPTSS	0	R/W	Setup command receive complete
2	EP0oTSS	0	R/W	EP0o receive complete
1	EP0iTRS	0	R/W	EP0i transfer request
0	EP0iTSS	0	R/W	EP0i transmit complete

24.3.13 USB Interrupt Select Register 2 (USBISR2)

USBISR2 selects the vector numbers of the interrupt requests indicated in the USB interrupt flag register 2 (USBIFR2). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR2 is cleared to 0, the interrupt will be USI0. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR2 is set to 1, the interrupt will be USI1. If interrupts occur simultaneously, USI0 has priority by default.

Bit:	7	6	5	4	3	2	1	0
	-	-	EP3TRS	EP3TSS	EP2TRS	EP2EMPTYS	EP2ALLEMPS	EP1FULLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP3TRS	0	R/W	EP3 transfer request
4	EP3TSS	0	R/W	EP3 transmit complete
3	EP2TRS	0	R/W	EP2 transfer request
2	EP2EMPTYS	0	R/W	EP2 FIFO empty
1	EP2ALLEMPS	0	R/W	EP2 FIFO all empty
0	EP1FULLS	0	R/W	EP1 FIFO full

24.3.14 USB Interrupt Select Register 3 (USBISR3)

USBISR3 selects the vector numbers of the interrupt requests indicated in the USB interrupt flag register 3 (USBIFR3). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR3 is cleared to 0, the interrupt will be USI0. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR3 is set to 1, the interrupt will be USI1. If interrupts occur simultaneously, USI0 has priority by default.

Bit:	7	6	5	4	3	2	1	0
	-	-	EP6TRS	EP6TSS	EP5TRS	EP5EMPTYS	EP5ALLEMPS	EP4FULLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP6TRS	0	R/W	EP6 transfer request
4	EP6TSS	0	R/W	EP6 transmit complete
3	EP5TRS	0	R/W	EP5 transfer request
2	EP5EMPTYS	0	R/W	EP5 FIFO empty
1	EP5ALLEMPS	0	R/W	EP5 FIFO all empty
0	EP4FULLS	0	R/W	EP4 FIFO full

24.3.15 USB Interrupt Select Register 4 (USBISR4)

USBISR4 selects the vector numbers of the interrupt requests indicated in the USB interrupt flag register 4 (USBIFR4). If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR4 is cleared to 0, the interrupt will be USI0. If the USB issues an interrupt request to the INTC when the corresponding bit in USBISR4 is set to 1, the interrupt will be USI1. If interrupts occur simultaneously, USI0 has priority by default.

Bit:	7	6	5	4	3	2	1	0
	-	-	EP9 TRS	EP9 TSS	EP8 TRS	EP8 EMPTYS	-	EP7 FULLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	EP9TRS	0	R/W	EP9 transfer request
4	EP9TSS	0	R/W	EP9 transmit complete
3	EP8TRS	0	R/W	EP8 transfer request
2	EP8EMPTYS	0	R/W	EP8 FIFO empty
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	EP7FULLS	0	R/W	EP7 FIFO full

24.3.16 USBEP0i Data Register (USBEPDR0i)

USBEPDR0i is a 16-byte transmit FIFO buffer for endpoint 0, holding one packet of transmit data for control IN. Transmit data is fixed by writing one packet of data and setting the EP0iPKTE bit in the USB trigger register 0 (USBTRG0). When an ACK handshake is returned from the host after the data has been transmitted, the EP0iTS bit in the USB interrupt flag register 1 (USBIFR1) is set to 1. USBEPDR0i can be initialized by the EP0iCLR bit in the USB FIFO clear register 0 (USBFCLR0). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for control IN transfer

24.3.17 USBEP0o Data Register (USBEPDR0o)

USBEPDR0o is a 16-byte receive FIFO buffer for endpoint 0 to store endpoint 0 receive data other than setup commands. When data is received normally, the EP0oTS bit in the USB interrupt flag register 1 (USBIFR1) is set, and the number of receive bytes is indicated in the USBEP0o receive data size register (USBEPSZ0o). After the data has been read, setting the EP0oRDFN bit in the USB trigger register 0 (USBTRG0) enables the next packet to be received. USBEPDR0o can be initialized by the EP0oCLR bit in the USB FIFO clear register 0 (USBFCLR0).

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	Data register for control OUT transfer

24.3.18 USBEP0s Data Register (USBEPDR0s)

USBEPDR0s is an 8-byte FIFO buffer specifically for receiving endpoint 0 setup commands. USBEPDR0s receives only setup commands requiring processing on the application side. When a command that this module automatically processes is received, it is not stored. When command data is stored normally, the SETUPTS bit in the USB interrupt flag register 1 (USBIFR1) is set.

As a setup command must be received without fail, if data is left in this buffer, it will be overwritten with new data. If reception of the next command is started while the current command is being read, command reception has priority and data read by the application is forcibly disabled. Therefore the read data is invalid.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	Register for storing the setup command on control OUT transfer

24.3.19 USBEP1 Data Register (USBEPDR1)

USBEPDR1 is a 128-byte receive FIFO buffer for endpoint 1. USBEPDR1 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When one packet of data is received normally from the host, the EP1FULL bit in the USB interrupt flag register 2 (USBIFR2) is set. The number of receive bytes is indicated in the USB EP1 receive data size register (USBEPSZ1). After the data has been read, the buffer that was read is enabled to receive again by writing 1 to the EP1RDFN bit in the USB trigger register 1 (USBTRG1). The receive data in this FIFO buffer can be transferred by DMA or DTC (byte-by-byte dual-address transfer). USBEPDR1 can be initialized by the EP1CLR bit in the USB FIFO clear register 1 (USBFCLR1).

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	Data register for endpoint 1 transfer

24.3.20 USBEP2 Data Register (USBEPDR2)

USBEPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. USBEPDR2 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and the EP2PKTE bit in the USB trigger register 1 (USBTRG1) is set, one packet of transmit data is fixed, and the dual buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA or DTC (byte-by-byte dual-address transfer). USBEPDR2 can be initialized by the EP2CLR bit in the USB FIFO clear register 1 (USBFCLR1). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for endpoint 2 transfer

24.3.21 USBEP3 Data Register (USBEPDR3)

USBEPDR3 is a 16-byte transmit FIFO buffer for endpoint 3, holding one packet of transmit data in endpoint 3 interrupt transfer. Transmit data is fixed by writing one packet of data and setting the EP3PKTE bit in the USB trigger register 1 (USBTRG1). USBEPDR3 can be initialized by the EP3CLR bit in the USB FIFO clear register 1 (USBFCLR1). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for endpoint 3 transfer

24.3.22 USBEP4 Data Register (USBEPDR4)

USBEPDR4 is a 128-byte receive FIFO buffer for endpoint 4. USBEPDR1 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When one packet of data is received normally from the host, the EP4FULL bit in the USB interrupt flag register 3 (USBIFR3) is set. The number of receive bytes is indicated in the USB EP4 receive data size register (USBEPSZ4). After the data has been read, the buffer that was read is enabled to receive again by writing 1 to the EP4RDFN bit in the USB trigger register 2 (USBTRG2). The receive data in this FIFO buffer can be transferred by DMA or DTC (byte-by-byte dual-address transfer). USBEPDR4 can be initialized by the EP4CLR bit in the USB FIFO clear register 2 (USBFCLR2).

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	Data register for endpoint 4 transfer

24.3.23 USBEP5 Data Register (USBEPDR5)

USBEPDR5 is a 128-byte transmit FIFO buffer for endpoint 5. USBEPDR5 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and the EP5PKTE bit in the USB trigger register 2 (USBTRG2) is set, one packet of transmit data is fixed, and the dual buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA or DTC (byte-by-byte dual-address transfer). USBEPDR5 can be initialized by the EP5CLR bit in the USB FIFO clear register 2 (USBFCLR2). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for endpoint 5 transfer

24.3.24 USBEP6 Data Register (USBEPDR6)

USBEPDR6 is a 16-byte transmit FIFO buffer for endpoint 6, holding one packet of transmit data in endpoint 6 interrupt transfer. Transmit data is fixed by writing one packet of data and setting the EP6PKTE bit in the USB trigger register 2 (USBTRG2). USBEPDR6 can be initialized by the EP6CLR bit in the USB FIFO clear register 2 (USBFCLR2). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for endpoint 6 transfer

24.3.25 USBEP7 Data Register (USBEPDR7)

USBEPDR7 is a 64-byte receive FIFO buffer for endpoint 7. When one packet of data is received normally from the host, the EP7FULL bit in the USB interrupt flag register 4 (USBIFR4) is set. The number of receive bytes is indicated in the USB EP7 receive data size register (USBEPSZ7). After the data has been read, the buffer that was read is enabled to receive again by writing 1 to the EP7RDFN bit in the USB trigger register 3 (USBTRG3). USBEPDR7 can be initialized by the EP7CLR bit in the USB FIFO clear register 3 (USBFCLR3).

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	R	Data register for endpoint 7 transfer

24.3.26 USBEP8 Data Register (USBEPDR8)

USBEPDR8 is a 64-byte transmit FIFO buffer for endpoint 8. Transmit data for one packet is fixed by writing transmit data to this FIFO buffer and setting the EP8PKTE bit in the USB trigger register 3 (USBTRG3). USBEPDR8 can be initialized by the EP8CLR bit in the USB FIFO clear register 3 (USBFCLR3). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for endpoint 8 transfer

24.3.27 USBEP9 Data Register (USBEPDR9)

USBEPDR9 is a 16-byte transmit FIFO buffer for endpoint 9, holding one packet of transmit data in endpoint 9 interrupt transfer. Transmit data is fixed by writing one packet of data and setting the EP9PKTE bit in the USB trigger register 3 (USBTRG3). USBEPDR9 can be initialized by the EP9CLR bit in the USB FIFO clear register 3 (USBFCLR3). The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Data register for endpoint 9 transfer

24.3.28 USBEP0o Receive Data Size Register (USBEPSZ0o)

USBEPSZ0o indicates, in bytes, the amount of data received from the host by endpoint 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	D4	D3	D2	D1	D0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0.
4 to 0	D4 to D0	All 0	R	Number of bytes received by endpoint 0

24.3.29 USBEP1 Receive Data Size Register (USBEPSZ1)

USBEPSZ1 indicates, in bytes, the amount of data received from the host by endpoint 1. The endpoint 1 FIFO buffer has a dual-FIFO configuration. This register indicates the receive data size of the currently selected FIFO (that can be read by CPU).

Bit:	7	6	5	4	3	2	1	0
	-	D6	D5	D4	D3	D2	D1	D0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	All 0	R	Reserved This bit is always read as 0.
6 to 0	D6 to D0	All 0	R	Number of bytes received by endpoint 1

24.3.30 USBEP4 Receive Data Size Register (USBEPSZ4)

USBEPSZ4 indicates, in bytes, the amount of data received from the host by endpoint 4. The endpoint 4 FIFO buffer has a dual-FIFO configuration. This register indicates the receive data size of the currently selected FIFO (that can be read by CPU).

Bit:	7	6	5	4	3	2	1	0
	-	D6	D5	D4	D3	D2	D1	D0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	All 0	R	Reserved This bit is always read as 0.
6 to 0	D6 to D0	All 0	R	Number of bytes received by endpoint 4

24.3.31 USBEP7 Receive Data Size Register (USBEPSZ7)

USBEPSZ7 indicates, in bytes, the amount of data received from the host by endpoint 7.

Bit:	7	6	5	4	3	2	1	0
	-	D6	D5	D4	D3	D2	D1	D0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	All 0	R	Reserved This bit is always read as 0.
6 to 0	D6 to D0	All 0	R	Number of bytes received by endpoint 7

24.3.32 USB Data Status Register 0 (USBDASTS0)

USBDASTS0 indicates whether the transmit FIFO buffer contains valid data. The EP0iDE bit is set to 1 when data is written to the corresponding FIFO buffer and the packet enable state is set. This bit is cleared when data has been completely transmitted to the host.

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	EP0iDE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0.
0	EP0iDE	0	R	EP0i Data Present This bit is set to 1 when the endpoint 0i FIFO buffer contains valid data.

24.3.33 USB Data Status Register 1 (USBDASTS1)

USBDASTS1 indicates whether the transmit FIFO buffer contains valid data. The EP2DE or EP3DE bit is set to 1 when data is written to the corresponding FIFO buffer and the packet enable state is set. This bit is cleared when data has been completely transmitted to the host.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP3DE	EP2DE	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0.
2	EP3DE	0	R	EP3 Data Present This bit is set to 1 when the endpoint 3 FIFO buffer contains valid data.
1	EP2DE	0	R	EP2 Data Present This bit is set to 1 when the endpoint 2 FIFO buffer contains valid data.
0	—	0	R	Reserved This bit is always read as 0.

24.3.34 USB Data Status Register 2 (USBDASTS2)

USBDASTS2 indicates whether the transmit FIFO buffer contains valid data. The EP5DE or EP6DE bit is set to 1 when data is written to the corresponding FIFO buffer and the packet enable state is set. This bit is cleared when data has been completely transmitted to the host.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP6DE	EP5DE	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0.
2	EP6DE	0	R	EP6 Data Present This bit is set to 1 when the endpoint 6 FIFO buffer contains valid data.
1	EP5DE	0	R	EP5 Data Present This bit is set to 1 when the endpoint 5 FIFO buffer contains valid data.
0	—	0	R	Reserved This bit is always read as 0.

24.3.35 USB Data Status Register 3 (USBDASTS3)

USBDASTS3 indicates whether the transmit FIFO buffer contains valid data. The EP8DE or EP9DE bit is set to 1 when data is written to the corresponding FIFO buffer and the packet enable state is set. This bit is cleared when data has been completely transmitted to the host.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP9DE	EP8DE	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	EP9DE	0	R	EP9 Data Present This bit is set to 1 when the endpoint 9 FIFO buffer contains valid data.
1	EP8DE	0	R	EP8 Data Present This bit is set to 1 when the endpoint 8 FIFO buffer contains valid data.
0	—	0	R	Reserved This bit is always read as 0.

24.3.36 USB Trigger Register 0 (USBTRG0)

USBTRG0 is a write-only register that generates one-shot triggers to control the transmit/receive sequence for endpoint 0. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP0s RDFN	EP0o RDFN	EP0i PKTE
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP0sRDFN	0	W	EP0s Read Complete Write 1 to this bit after EP0s command FIFO data has been read. Writing 1 to this bit enables transmission/reception of data in the following data stage. A NACK handshake is returned in response to transmit/receive requests from the host in the data stage until 1 is written to this bit.
1	EP0oRDFN	0	W	EP0o Read Complete Writing 1 to this bit after one packet of data has been read from the endpoint 0 receive FIFO buffer initializes the FIFO buffer, enabling the next packet to be received.
0	EP0iPKTE	0	W	EP0i Packet Enable After one packet of data has been written to the endpoint 0 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.

24.3.37 USB Trigger Register 1 (USBTRG1)

USBTRG1 is a write-only register that generates one-shot triggers to control the transmit/receive sequence for each endpoint. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP3 PKTE	EP2 PKTE	EP1 RDFN
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP3PKTE	0	W	EP3 Packet Enable After one packet of data has been written to the endpoint 3 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
1	EP2PKTE	0	W	EP2 Packet Enable After one packet of data has been written to the endpoint 2 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
0	EP1RDFN	0	W	EP1 Read Complete Write 1 to this bit after one packet of data has been read from the endpoint 1 receive FIFO buffer. The endpoint 1 receive FIFO buffer has a dual-FIFO configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.

24.3.38 USB Trigger Register 2 (USBTRG2)

USBTRG2 is a write-only register that generates one-shot triggers to control the transmit/receive sequence for each endpoint. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP6 PKTE	EP5 PKTE	EP4 RDFN
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP6PKTE	0	W	EP6 Packet Enable After one packet of data has been written to the endpoint 6 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
1	EP5PKTE	0	W	EP5 Packet Enable After one packet of data has been written to the endpoint 5 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
0	EP4RDFN	0	W	EP4 Read Complete Write 1 to this bit after one packet of data has been read from the endpoint 4 receive FIFO buffer. The endpoint 4 receive FIFO buffer has a dual-FIFO configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.

24.3.39 USB Trigger Register 3 (USBTRG3)

USBTRG3 generates one-shot triggers to control the transmit/receive sequence for each endpoint.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP6 PKTE	EP5 PKTE	EP4 RDFN
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP9PKTE	0	W	EP9 Packet Enable After one packet of data has been written to the endpoint 9 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
1	EP8PKTE	0	W	EP8 Packet Enable After one packet of data has been written to the endpoint 8 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
0	EP7RDFN	0	W	EP7 Read Complete Write 1 to this bit after one packet of data has been read from the endpoint 7 receive FIFO buffer. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.

24.3.40 USB FIFO Clear Register 0 (USBFCLR0)

USBFCLR0 is a write-only register to initialize the FIFO buffers for endpoint 0. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear the FIFO buffer during transmission/reception. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	EP0o CLR	EP0i CLR
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	-	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	—	Reserved The write value should always be 0.
1	EP0oCLR	0	W	EP0o Clear Writing 1 to this bit initializes the endpoint 0 receive FIFO buffer.
0	EP0iCLR	0	W	EP0i Clear Writing 1 to this bit initializes the endpoint 0 transmit FIFO buffer.

24.3.41 USB FIFO Clear Register 1 (USBFCLR1)

USBFCLR1 is a write-only register to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear the FIFO buffer during transmission/reception. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP3 CLR	EP2 CLR	EP1 CLR
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP3CLR	0	W	EP3 Clear Writing 1 to this bit initializes the endpoint 3 transmit FIFO buffer.
1	EP2CLR	0	W	EP2 Clear Writing 1 to this bit initializes both endpoint 2 transmit FIFO buffers.
0	EP1CLR	0	W	EP1 Clear Writing 1 to this bit initializes both endpoint 1 receive FIFO buffers.

24.3.42 USB FIFO Clear Register 2 (USBFCLR2)

USBFCLR2 is a write-only register to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear the FIFO buffer during transmission/reception. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP6 CLR	EP5 CLR	EP4 CLR
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP6CLR	0	W	EP6 Clear Writing 1 to this bit initializes the endpoint 6 transmit FIFO buffer.
1	EP5CLR	0	W	EP5 Clear Writing 1 to this bit initializes both endpoint 5 transmit FIFO buffers.
0	EP4CLR	0	W	EP4 Clear Writing 1 to this bit initializes both endpoint 4 receive FIFO buffers.

24.3.43 USB FIFO Clear Register 3 (USBFCLR3)

USBFCLR3 is a write-only register to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear the FIFO buffer during transmission/reception. The read value of this register is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	EP9CLR	EP8CLR	EP7CLR
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	-	-	-	-	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	—	Reserved The write value should always be 0.
2	EP9CLR	0	W	EP9 Clear Writing 1 to this bit initializes the endpoint 9 transmit FIFO buffer.
1	EP8CLR	0	W	EP8 Clear Writing 1 to this bit initializes the endpoint 8 transmit FIFO buffer.
0	EP7CLR	0	W	EP7 Clear Writing 1 to this bit initializes the endpoint 7 receive FIFO buffer.

24.3.44 USB Endpoint Stall Register 0 (USBEPSTL0)

The bits in USBEPSTL0 are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host.

The EP0STLC bit is used to clear the EP0STLS stall setting. The EP0STLC and EP0STLS bits must not be set to 1 simultaneously.

The stall bit for endpoint 0 (EP0STLS) is cleared automatically on reception of 8-bit command data to be decoded in this function module. When the SETUPTS flag in USBIFR1 is set, writing 1 to the EP0STLS bit is ignored. For details, see section 24.7, Stall Operations.

USBEPSTL0 contains a write-only bit. The read value of such a bit is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	EP0 STLC	-	-	-	EP0 STLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	EP0STLC	0	W	EP0 Stall Clear Writing 1 to this bit clears the EP0STLS bit to 0. This bit cannot be cleared to 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	EP0STLS	0	R/W	EP0 Stall Setting Writing 1 to this bit places endpoint 0 in the stall state. This bit cannot be cleared to 0.

24.3.45 USB Endpoint Stall Register 1 (USBEPSTL1)

The bits in USBEPSTL1 are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. Bits EP1STLC to EP3STLC are used to clear bits EP1STLS to EP3STLS. The stall setting bit and stall clear bit for the same endpoint must not be set to 1 simultaneously. For details, see section 24.7, Stall Operations.

USBEPSTL1 contains a write-only bit. The read value of such a bit is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	EP3 STLC	EP2 STLC	EP1 STLC	-	EP3 STLS	EP2 STLS	EP1 STLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	-	W	W	W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP3STLC	0	W	EP3 Stall Clear Writing 1 to this bit clears the EP3STLS bit to 0. This bit cannot be cleared to 0.
5	EP2STLC	0	W	EP2 Stall Clear Writing 1 to this bit clears the EP2STLS bit to 0. This bit cannot be cleared to 0.
4	EP1STLC	0	W	EP1 Stall Clear Writing 1 to this bit clears the EP1STLS bit to 0. This bit cannot be cleared to 0.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	EP3STLS	0	R/W	EP3 Stall Setting Writing 1 to this bit places endpoint 3 in the stall state. This bit cannot be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
1	EP2STLS	0	R/W	EP2 Stall Setting Writing 1 to this bit places endpoint 2 in the stall state. This bit cannot be cleared to 0.
0	EP1STLS	0	R/W	EP1 Stall Setting Writing 1 to this bit places endpoint 1 in the stall state. This bit cannot be cleared to 0.

24.3.46 USB Endpoint Stall Register 2 (USBEPSTL2)

The bits in USBEPSTL2 are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. Bits EP4STLC to EP6STLC are used to clear bits EP4STLS to EP6STLS. The stall setting bit and stall clear bit for the same endpoint must not be set to 1 simultaneously. For details, see section 24.7, Stall Operations.

USBEPSTL2 contains a write-only bit. The read value of such a bit is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	EP6 STLC	EP5 STLC	EP4 STLC	-	EP6 STLS	EP5 STLS	EP4 STLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	W	W	W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP6STLC	0	W	EP6 Stall Clear Writing 1 to this bit clears the EP6STLS bit to 0. This bit cannot be cleared to 0.
5	EP5STLC	0	W	EP5 Stall Clear Writing 1 to this bit clears the EP5STLS bit to 0. This bit cannot be cleared to 0.
4	EP4STLC	0	W	EP4 Stall Clear Writing 1 to this bit clears the EP4STLS bit to 0. This bit cannot be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	EP6STLS	0	R/W	EP6 Stall Setting Writing 1 to this bit places endpoint 6 in the stall state. This bit cannot be cleared to 0.
1	EP5STLS	0	R/W	EP5 Stall Setting Writing 1 to this bit places endpoint 5 in the stall state. This bit cannot be cleared to 0.
0	EP4STLS	0	R/W	EP4 Stall Setting Writing 1 to this bit places endpoint 4 in the stall state. This bit cannot be cleared to 0.

24.3.47 USB Endpoint Stall Register 3 (USBEPSTL3)

The bits in USBEPSTL3 are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. Bits EP7STLC to EP9STLC are used to clear bits EP7STLS to EP9STLS. The stall setting bit and stall clear bit for the same endpoint must not be set to 1 simultaneously. For details, see section 24.7, Stall Operations.

USBEPSTL3 contains a write-only bit. The read value of such a bit is undefined. Do not write a value to this register using the read value, such as a bit manipulation instruction.

Bit:	7	6	5	4	3	2	1	0
	-	EP9 STLC	EP8 STLC	EP7 STLC	-	EP9 STLS	EP8 STLS	EP7 STLS
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	W	W	W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP9STLC	0	W	EP9 Stall Clear Writing 1 to this bit clears the EP9STLS bit to 0. This bit cannot be cleared to 0.

Bit	Bit Name	Initial Value	R/W	Description
5	EP8STLC	0	W	EP8 Stall Clear Writing 1 to this bit clears the EP8STLS bit to 0. This bit cannot be cleared to 0.
4	EP7STLC	0	W	EP7 Stall Clear Writing 1 to this bit clears the EP7STLS bit to 0. This bit cannot be cleared to 0.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	EP9STLS	0	R/W	EP9 Stall Setting Writing 1 to this bit places endpoint 9 in the stall state. This bit cannot be cleared to 0.
1	EP8STLS	0	R/W	EP8 Stall Setting Writing 1 to this bit places endpoint 8 in the stall state. This bit cannot be cleared to 0.
0	EP7STLS	0	R/W	EP7 Stall Setting Writing 1 to this bit places endpoint 7 in the stall state. This bit cannot be cleared to 0.

24.3.48 USB Stall Status Register 1 (USBSTLSR1)

Bits 0 to 2 in USBSTLSR1 indicate the internal stall status of endpoints 1 to 3. A value 1 shows stall status, and 0 shows normal status. These bits are status bits, and cannot be cleared. Bits 4 to 6 in USBSTLSR1 are automatic stall clear enable bits for endpoints 1 to 3.

Bit:	7	6	5	4	3	2	1	0
	-	EP3 ASCE	EP2 ASCE	EP1 ASCE	-	EP3 STLST	EP2 STLST	EP1 STLST
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP3ASCE	0	R/W	EP3 Automatic Stall Clear Enable When the EP3ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP3 stall setting bit (EP3STLS in USBEPSTL1) is automatically cleared to 0. When EP3ASCE = 0, the EP3STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP3ASCE = 1 before setting the EP3STLS bit in USBEPSTL1 to 1.
5	EP2ASCE	0	R/W	EP2 Automatic Stall Clear Enable When the EP2ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP2 stall setting bit (EP2STLS in USBEPSTL1) is automatically cleared to 0. When EP2ASCE = 0, the EP2STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP2ASCE = 1 before setting the EP2STLS bit in USBEPSTL1 to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	EP1ASCE	0	R/W	EP1 Automatic Stall Clear Enable When the EP1ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP1 stall setting bit (EP1STLS in USBEPSTL1) is automatically cleared to 0. When EP1ASCE = 0, the EP1STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP1ASCE = 1 before setting the EP1STLS bit in USBEPSTL1 to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	EP3STLST	0	R	EP3 Internal Stall Status
1	EP2STLST	0	R	EP2 Internal Stall Status
0	EP1STLST	0	R	EP1 Internal Stall Status

24.3.49 USB Stall Status Register 2 (USBSTLSR2)

Bits 0 to 2 in USBSTLSR2 indicate the internal stall status of endpoints 4 to 6. A value 1 shows stall status, and 0 shows normal status. These bits are status bits, and cannot be cleared. Bits 4 to 6 in USBSTLSR2 are automatic stall clear enable bits for endpoints 4 to 6.

Bit:	7	6	5	4	3	2	1	0
	-	EP3 ASCE	EP2 ASCE	EP1 ASCE	-	EP6 STLST	EP5 STLST	EP4 STLST
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP6ASCE	0	R/W	EP6 Automatic Stall Clear Enable When the EP6ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP6 stall setting bit (EP6STLS in USBEPSTL2) is automatically cleared to 0. When EP6ASCE = 0, the EP6STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP6ASCE = 1 before setting the EP6STLS bit in USBEPSTL2 to 1.
5	EP5ASCE	0	R/W	EP5 Automatic Stall Clear Enable When the EP5ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP5 stall setting bit (EP5STLS in USBEPSTL2) is automatically cleared to 0. When EP5ASCE = 0, the EP5STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP5ASCE = 1 before setting the EP5STLS bit in USBEPSTL2 to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	EP4ASCE	0	R/W	EP4 Automatic Stall Clear Enable When the EP4ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP4 stall setting bit (EP4STLS in USBEPSTL2) is automatically cleared to 0. When EP4ASCE = 0, the EP4STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP4ASCE = 1 before setting the EP4STLS bit in USBEPSTL2 to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	EP6STLST	0	R	EP6 Internal Stall Status
1	EP5STLST	0	R	EP5 Internal Stall Status
0	EP4STLST	0	R	EP4 Internal Stall Status

24.3.50 USB Stall Status Register 3 (USBSTLSR3)

Bits 0 to 2 in USBSTLSR3 indicate the internal stall status of endpoints 4 to 6. A value 1 shows stall status, and 0 shows normal status. These bits are status bits, and cannot be cleared. Bits 4 to 6 in USBSTLSR3 are automatic stall clear enable bits for endpoints 4 to 6.

Bit:	7	6	5	4	3	2	1	0
	-	EP9 ASCE	EP8 ASCE	EP7 ASCE	-	EP9 STLST	EP8 STLST	EP7 STLST
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	EP9ASCE	0	R/W	EP9 Automatic Stall Clear Enable When the EP9ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP9 stall setting bit (EP9STLS in USBEPSTL3) is automatically cleared to 0. When EP9ASCE = 0, the EP9STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP9ASCE = 1 before setting the EP9STLS bit in USBEPSTL3 to 1.
5	EP8ASCE	0	R/W	EP8 Automatic Stall Clear Enable When the EP8ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP5 stall setting bit (EP8STLS in USBEPSTL3) is automatically cleared to 0. When EP8ASCE = 0, the EP8STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP8ASCE = 1 before setting the EP8STLS bit in USBEPSTL3 to 1.

Bit	Bit Name	Initial Value	R/W	Description
4	EP7ASCE	0	R/W	EP7 Automatic Stall Clear Enable When the EP7ASCE bit is set to 1, a stall handshake is returned to the host, and then the EP7 stall setting bit (EP7STLS in USBEPSTL3) is automatically cleared to 0. When EP7ASCE = 0, the EP7STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP7ASCE = 1 before setting the EP7STLS bit in USBEPSTL3 to 1.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	EP9STLST	0	R	EP9 Internal Stall Status
1	EP8STLST	0	R	EP8 Internal Stall Status
0	EP7STLST	0	R	EP7 Internal Stall Status

24.3.51 USB DMA Transfer Setting Register (USBDMAR)

USBDMAR enables DMA or DTC transfer between the data registers for endpoints 1, 2, 4 and 5 and the memory by the on-chip direct memory access controller (DMAC) or on-chip data transfer controller (DTC). Dual-address transfer on a per-byte basis is performed. To start DMA transfer, DMAC settings must be made in addition to the settings in this register. For details of DMA transfer, see section 24.8, DMA Transfer. To start DTC transfer, DTC settings must be made in addition to the settings in this register. For details of DTC transfer, see section 24.9, DTC Transfer.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	EP5 DMAE*	EP4 DMAE*	-	EP2 DMAE*	EP1 DMAE*
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	EP5DMAE*	0	R/W	EP5 DMA/DTC Transfer Enable When this bit is set, DMA/DTC transfer is enabled from the memory to the endpoint 5 transmit FIFO buffer. If there is at least one byte of space in the FIFO buffer, a transfer request is asserted to the DMAC or DTC. During DMA/DTC transfer, when 64 bytes are written to the FIFO buffer, the EP5 packet enable bit is set automatically, allowing 64 bytes of data to be transferred. If there is still space in the other of the two FIFO buffers, a transfer request is asserted to the DMAC or DTC again. However, if the size of the data packet to be transmitted is less than 64 bytes, the EP5 packet enable bit is not set automatically, and so should be set by the CPU with a DMA/DTC transfer end interrupt. Also, as EP5-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the USB interrupt enable register.

Bit	Bit Name	Initial Value	R/W	Description
3	EP4DMAE*	0	R/W	EP4 DMA/DTC Transfer Enable When this bit is set, DMA/DTC transfer is enabled from the endpoint 4 receive FIFO buffer to the memory. If at least one byte of receive data is remaining in the FIFO buffer, a transfer request is asserted to the DMAC or DTC. During DMA/DTC transfer, when all the received data is read, an EP4 read completion trigger is given. Also, as EP4-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the USB interrupt enable register.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	EP2DMAE*	All 0	R	EP2 DMA/DTC Transfer Enable When this bit is set, DMA/DTC transfer is enabled from the memory to the endpoint 2 transmit FIFO buffer. If there is at least one byte of space in the FIFO buffer, a transfer request is asserted to the DMAC or DTC. During DMA/DTC transfer, when 64 bytes are written to the FIFO buffer, the EP2 packet enable bit is set automatically, allowing 64 bytes of data to be transferred. If there is still space in the other of the two FIFO buffers, a transfer request is asserted to the DMAC or DTC again. However, if the size of the data packet to be transmitted is less than 64 bytes, the EP2 packet enable bit is not set automatically, and so should be set by the CPU with a DMA/DTC transfer end interrupt. Also, as EP2-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the USB interrupt enable register.

Bit	Bit Name	Initial Value	R/W	Description
0	EP5DMAE*	0	R/W	EP1 DMA/DTC Transfer Enable When this bit is set, DMA/DTC transfer is enabled from the endpoint 1 receive FIFO buffer to the memory. If at least one byte of receive data is remaining in the FIFO buffer, a transfer request is asserted to the DMAC or DTC. During DMA/DTC transfer, when all the received data is read, an EP1 read completion trigger is given. Also, as EP1-related interrupt requests to the CPU are not automatically masked, interrupt requests should be masked as necessary in the USB interrupt enable register.

Note: * To start DMA transfer, set the DME bit in DMAOR before setting this bit.
To start DTC transfer, set the corresponding DTCE bit in DTCER before setting this bit.

24.3.52 USB Configuration Value Register (USBCVR)

USBCVR stores Configuration Setting, Interface Setting, or Alternate Setting value that is set when the Set Configuration or Set Interface command is received successfully.

Bit:	7	6	5	4	3	2	1	0
	CNFV1	CNFV0	INTV1	INTV0	-	ALTV2	ALTV1	ALTV0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	CNFV1	0	R	These bits store the Configuration Setting value when the Set Configuration command is received. The CNFV value is updated when the SETC bit in USBIFR0 is set to 1.
6	CNFV0	0	R	
5	INTV1	0	R	These bits store the Interface Setting value when the Set Interface command is received. The INTV value is updated when the SETI bit in USBIFR0 is set to 1.
4	INTV0	0	R	
3	—	0	R	Reserved This bit is always read as 0.
2	ALTV2	0	R	These bits store the Alternate Setting value when the Set Interface command is received. The ALTV value is updated when the SETI bit in USBIFR0 is set to 1.
1	ALTV1	0	R	
0	ALTV0	0	R	

24.3.53 USB Control Register (USBCTLR)

USBCTLR is used to set functions for PRTRST and ASCE.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	EP0 ASCE	PRTRST
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 0	R	Reserved The write value should always be 0.
1	EP0ASCE	0	R/W	EP0 Automatic Stall Clear Enable When EP0ASCE is set to 1, a stall handshake is returned to the host, and then the EP0 stall setting bit (EP0STLS in USBEPSTL0) is automatically cleared to 0. When EP0ASCE = 0, the EP0STLS bit is not automatically cleared to 0 and must be cleared by the user. To enable this bit, be sure to set EP0ASCE = 1 before setting the EP0STLS bit in USBEPSTL0 to 1.
0	PRTRST	1	R/W	Protocol Processor Reset 0: The protocol processor is set to the active state. 1: The protocol processor is set to the reset state.

24.3.54 USB Endpoint Information Register (USBEPIR)

USBEPIR is used to set information of each endpoint, requiring five bytes per endpoint. Perform data write to this register sequentially beginning with logical endpoint 0. The total amount of write data should be within 50 bytes (5 bytes x 10 endpoints). Write endpoint information to this register once at a power-on reset, and do not write after that. Write data for an endpoint is described below.

Although there is one USBEPIR as data is written sequentially at the same address, the write data for endpoint 0 is shown as USBEPIR00 to USBEPIR04 (USBEPIR [endpoint number] [writing order]) for convenience of explanation. Write data to this register sequentially beginning with USBEPIR00. The read value is undefined.

Bit:	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

- USBEPIR00

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	D7 to D4	—	W	Endpoint Number [Settable range] 0 to 9
3, 2	D3, D2	—	W	Configuration Number Containing Endpoint [Settable range] 0 or 1
1, 0	D1, D0	—	W	Interface Number Containing Endpoint [Settable range] 0 to 3

- USBEPIR01

Bit	Bit Name	Initial Value	R/W	Description
7, 6	D7, D6	—	W	Alternate Number Containing Endpoint Fix these values to 0.
5, 4	D5, D4	—	W	Endpoint Transfer Method [Settable range] 0: Control 1: Setting prohibited 2: Bulk 3: Interrupt
3	D3	—	W	Endpoint Transfer Direction [Settable range] 0: Out 1: In
2 to 0	D2 to D0	—	W	Reserved [Settable range] 0: Fixed

- USBEPIR02

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	D7 to D1	—	W	Maximum Packet Size for Endpoint [Settable range] 0 to 64
0	D0	—	W	Reserved [Settable range] 0: Fixed

- USBEPIR03

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Reserved [Settable range] 0: Fixed

- USBEPIR04

Bit	Bit Name	Initial Value	R/W	Description
7 to 0	D7 to D0	—	W	Endpoint FIFO Number [Settable range] 0 to 9

Endpoint numbers are used by the USB host. Endpoint FIFO numbers correspond to endpoint numbers appearing in this manual. Therefore, making endpoint numbers correspond to endpoint FIFO numbers one-to-one with this information allows data transfer between USB host and endpoint FIFO. However, note the following restrictions for settings.

Each endpoint FIFO is optimized by the dedicated hardware that meets the transfer method, transfer direction, and maximum packet size. Therefore, be sure to observe the settings for transfer method, transfer direction, and maximum packet size shown in table 24.3.

1. Ensure that endpoint 0 corresponds to endpoint FIFO number 0.
2. The maximum packet size for endpoint FIFO number 0 can be set to 16 only.
3. Only maximum packet size can be set for endpoint FIFO number 0, and the other bits are all 0.
4. The maximum packet size for endpoint FIFO numbers 1, 2, 4, 5, 7, and 8 can be set to 64 only.
5. Only "Bulk transfer" and "Out" can be set for endpoint FIFO numbers 1, 4, and 7.
6. Only "Bulk transfer" and "In" can be set for endpoint FIFO numbers 2, 5, and 8.
7. The maximum packet size for endpoint FIFO numbers 3, 6, and 9 can be set to 16 only.
8. Only "Interrupt transfer" and "In" can be set for endpoint FIFO numbers 3, 6, and 9.
9. Information for up to 10 endpoints can be set.
10. Information for 10 endpoints must be written.
11. Write all 0 for information of unused endpoints.

Table 24.3 lists the settable transfer method, transfer direction, and maximum packet size.

Table 24.3 Restrictions for Settings

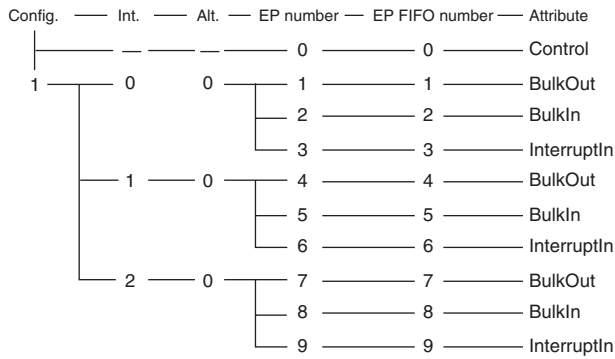
Endpoint FIFO Number	Maximum Packet Size	Transfer Method	Transfer Direction
0	16 bytes	Control	In/Out
1	64 bytes	Bulk	Out
2	64 bytes	Bulk	In
3	16 bytes	Interrupt	In
4	64 bytes	Bulk	Out
5	64 bytes	Bulk	In
6	16 bytes	Interrupt	In
7	64 bytes	Bulk	Out
8	64 bytes	Bulk	In
9	16 bytes	Interrupt	In

Table 24.4 shows a specific setting example.

Table 24.4 Setting Example

EP Number	Conf.	Int.	Alt.	Transfer Method	Transfer Direction	Maximum Packet Size	EP FIFO Number
0	—	—	—	Control	In/Out	16 bytes	0
1	1	0	0	Bulk	Out	64 bytes	1
2	1	0	0	Bulk	In	64 bytes	2
3	1	0	0	Interrupt	In	16 bytes	3
4	1	1	0	Bulk	Out	64 bytes	4
5	1	1	0	Bulk	In	64 bytes	5
6	1	1	0	Interrupt	In	16 bytes	6
7	1	2	0	Bulk	Out	64 bytes	7
8	1	2	0	Bulk	In	64 bytes	8
9	1	2	0	Interrupt	In	16 bytes	9

N	USBEPIR[N]0	USBEPIR[N]1	USBEPIR[N]2	USBEPIR[N]3	USBEPIR[N]4
0	00	00	20	00	00
1	14	20	80	00	01
2	24	28	80	00	02
3	34	38	20	00	03
4	45	20	80	00	04
5	55	28	80	00	05
6	65	38	20	00	06
7	76	20	80	00	07
8	86	28	80	00	08



24.3.55 USB Transceiver Test Register 0 (USBTRNTREG0)

USBTRNTREG0 is a test register that controls the on-chip transceiver output signals. Setting PTSTE = 1 enables the transceiver output signals (USD+, USD-) to be set arbitrarily. Table 24.5 shows the USBTRNTREG0 setting and pin output state.

Bit:	7	6	5	4	3	2	1	0
	PTSTE	-	-	-	SUS PEND	txenl	txse0	txdata
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PTSTE	0	R/W	Pin Test Enable Enables test control for the on-chip transceiver output pins (USD+/USD-).
6 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	SUSPEND	0	R/W	On-Chip Transceiver Output Signal Setting SUSPEND: Sets the on-chip transceiver suspend (SUSPEND) signal. txenl: Sets the on-chip transceiver output enable (txenl) signal. txse0: Sets the on-chip transceiver single-ended 0 (txse0) signal. txdata: Sets the on-chip transceiver data (txdata) signal.
2	txenl	0	R/W	
1	txse0	0	R/W	
0	txdata	0	R/W	

Table 24.5 USBTRNTREG0 Setting and Pin Output State

Pin Input	Register Setting				Pin Output State	
VBUS	PTSTE	txenl	txenl	txdata	USD+	USD-
0	×	×	×	×	Hi-Z	Hi-Z
1	0	×	×	×	—	—
1	1	0	0	0	0	1
1	1	0	0	1	1	0
1	1	0	1	×	0	0
1	1	1	×	×	Hi-Z	Hi-Z

[Legend]

×: Don't care

—: Uncontrollable pin state in normal operation, depending on the USB operating status and port settings.

24.3.56 USB Transceiver Test Register 1 (USBTRNTREG1)

USBTRNTREG1 is a test register that monitors the on-chip transceiver input signals. Setting PTSTE = 1 and txenl = 1 in USBTRNTREG0 enables monitoring of the transceiver input signals. Table 24.6 shows pin input values and monitored USBTRNTREG1 values.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	xver_data	dpls	dmns
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	xver_data	0	R	On-Chip Transceiver Input Signal Monitor
1	dpls	0	R	xver_data: Monitors the on-chip transceiver differential input level (xver_data) signal.
0	dmns	0	R	dpls: Monitors the on-chip transceiver USD+ (dpls) signal. dmns: Monitors the on-chip transceiver USD- (dmns) signal.

Table 24.6 Pin Input Values and Monitored USBTRNTREG1 Values

Register Setting		Pin Input Value		Monitored USBTRNTREG1 Value			
PTSTE	**SUSPEND**	**USD+**	**USD-**	**xver_data**	**dpls**	**dmns**	**Remarks**
0	×	×	×	0	0	0	Cannot be monitored when VBUS = 0 or PTSTE = 0.
1	0	0	0	×	0	0	Can be monitored when VBUS = 1 and PTSTE = 1.
1	0	0	1	0	0	1	
1	0	1	0	1	1	0	
1	0	1	1	×	1	1	
1	1	0	0	0	0	0	
1	1	0	1	0	0	1	
1	1	1	0	0	1	0	
1	1	1	1	0	1	1	

[Legend]

×: Don't care

24.4 Interrupt Sources

This module has six interrupt signals. Table 24.7 shows interrupt sources and their corresponding interrupt request signals. USI0, USI1, USBRXI0, USBTXI0, USBRXI1, and USBTXI1 interrupt signals are active low. The USBINTN interrupt is detected only by level.

Table 24.7 Interrupt Signals

Register	Bit	Transfer Mode	Interrupt Source	Description	Interrupt Request Signal	DMAC/DTC Activation
USBIFR0	0	Status	VBUSF	USB bus connection/disconnection detection	USI0 or USI1	×
	1		VBUSMN	VBUS connection status	—	×
	2		SETI	Set_Interface command detection	USI0 or USI1	×
	3		SETC	Set_Configuration command detection	USI0 or USI1	×
	4	—	Reserved	—	—	—
	5	—	Reserved	—	—	—
	6	Status	CFDN	Endpoint information loading complete	USI0 or USI1	×
	7		BRST	Bus reset	USI0 or USI1	×
USBIFR1	0	Control transfer (EP0)	EP0iTS*	EP0i transmit complete	USI0 or USI1	×
	1		EP0iTR*	EP0i transfer request	USI0 or USI1	×
	2		EP0oTS*	EP0o receive complete	USI0 or USI1	×
	3		SETUPTS*	Setup command receive complete	USI0 or USI1	×
	4	Status	SOF	SOF packet detection	USI0 or USI1	×
	5	—	Reserved	—	—	—
	6	—	Reserved	—	—	—
	7	—	Reserved	—	—	—

Register	Bit	Transfer Mode	Interrupt Source	Description	Interrupt Request Signal	DMAC/DTC Activation
USBIFR2	0	Bulk_out transfer (EP1)	EP1FULL	EP1FIFO full	USI0 or USI1	USBRXI0
	1	Bulk_in transfer (EP2)	EP2ALLEMP	EP2FIFO all empty	USI0 or USI1	×
	2		EP2EMPTY	EP2FIFO empty	USI0 or USI1	USBTXI0
	3		EP2TR	EP2 transfer request	USI0 or USI1	×
	4	Interrupt_in transfer (EP3)	EP3TS	EP3 transmit complete	USI0 or USI1	×
	5		EP3TR	EP3 transfer request	USI0 or USI1	×
	6	—	Reserved	—	—	—
	7	—	Reserved	—	—	—
USBIFR3	0	Bulk_out transfer (EP4)	EP4FULL	EP4FIFO full	USI0 or USI1	USBRXI1
	1	Bulk_in transfer (EP5)	EP5ALLEMP	EP5FIFO all empty	USI0 or USI1	×
	2		EP5EMPTY	EP5FIFO empty	USI0 or USI1	USBTXI1
	3		EP5TR	EP5 transfer request	USI0 or USI1	×
	4	Interrupt_in transfer (EP6)	EP6TS	EP6 transmit complete	USI0 or USI1	×
	5		EP6TR	EP6 transfer request	USI0 or USI1	×
	6	—	Reserved	—	—	—
	7	—	Reserved	—	—	—
USBIFR4	0	Bulk_out transfer (EP7)	EP7FULL	EP7FIFO full	USI0 or USI1	×
	1	—	Reserved	—	—	—
	2	Bulk_in transfer (EP8)	EP8EMPTY	EP8FIFO empty	USI0 or USI1	×
	3		EP8TR	EP8 transfer request	USI0 or USI1	×
	4	Interrupt_in transfer (EP9)	EP9TS	EP9 transmit complete	USI0 or USI1	×
	5		EP9TR	EP9 transfer request	USI0 or USI1	×
	6	—	Reserved	—	—	—
	7	—	Reserved	—	—	—

Note: * EP0-related interrupt sources must be assigned to the same interrupt request signal.

(1) USI0 signal

The USI0 signal requests interrupts from the sources for which the corresponding bits in the interrupt select register 0, 1, 2, 3 or 4 (any of USBISR0 to USBISR4) are cleared to 0. This signal is asserted if any USB interrupt flag register bit that corresponds to the interrupt source assigned to this signal is set to 1.

(2) USI1 signal

The USI1 signal requests interrupts from the sources for which the corresponding bits in the interrupt select register 0, 1, 2, 3 or 4 (any of USBISR0 to USBISR4) are set to 1. This signal is asserted if any USB interrupt flag register bit that corresponds to the interrupt source assigned to this signal is set to 1.

(3) USBRXI0 signal

USBRXI0 is a DMAC/DTC activation interrupt signal only for EP1. For details, see section 24.8, DMA Transfer and section 24.9, DTC Transfer.

(4) USBTXI0 signal

USBTXI0 is a DMAC/DTC activation interrupt signal only for EP2. For details, see section 24.8, DMA Transfer and section 24.9, DTC Transfer.

(5) USBRXI1 signal

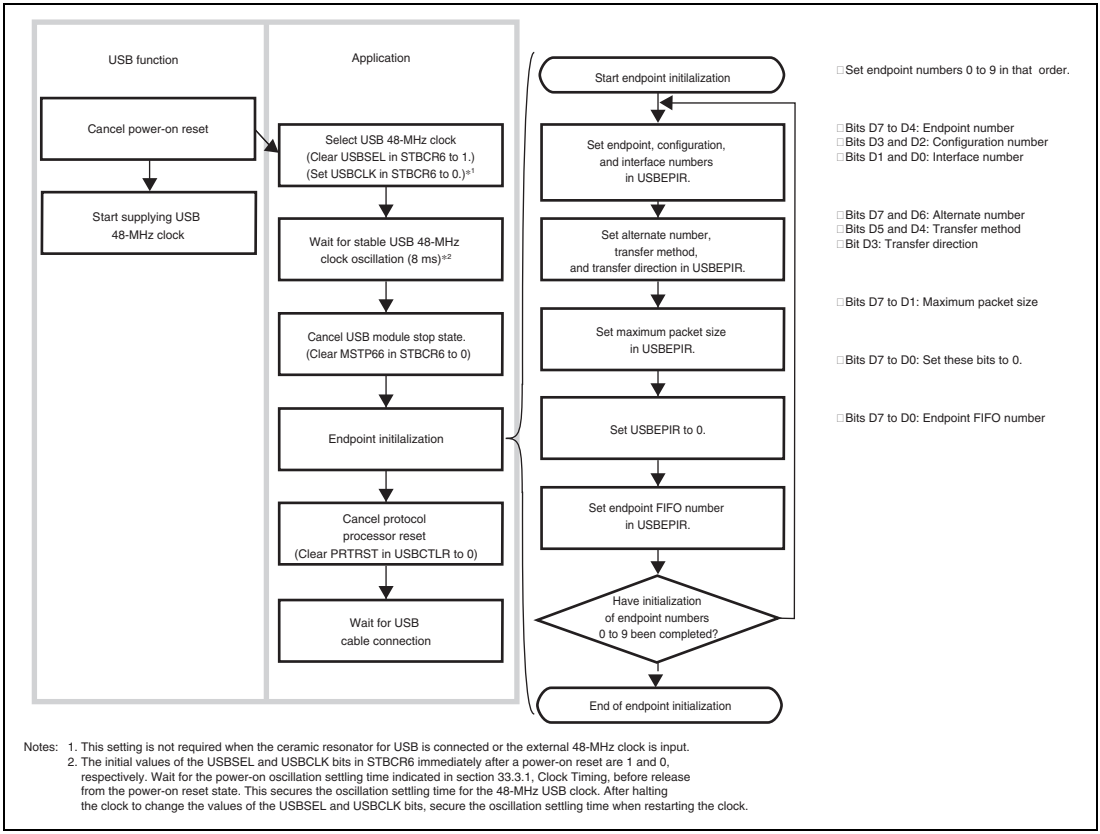
USBRXI1 is a DMAC/DTC activation interrupt signal only for EP4. For details, see section 24.8, DMA Transfer and section 24.9, DTC Transfer.

(6) USBTXI1 signal

USBTXI1 is a DMAC/DTC activation interrupt signal only for EP5. For details, see section 24.8, DMA Transfer and section 24.9, DTC Transfer.

24.5 Operation

24.5.1 Initial Settings



Notes: 1. This setting is not required when the ceramic resonator for USB is connected or the external 48-MHz clock is input.
2. The initial values of the USBSEL and USBCLK bits in STBCR6 immediately after a power-on reset are 1 and 0, respectively. Wait for the power-on oscillation settling time indicated in section 33.3.1, Clock Timing, before release from the power-on reset state. This secures the oscillation settling time for the 48-MHz USB clock. After halting the clock to change the values of the USBSEL and USBCLK bits, secure the oscillation settling time when restarting the clock.

Figure 24.2 Initial Setting

24.5.2 Cable Connection

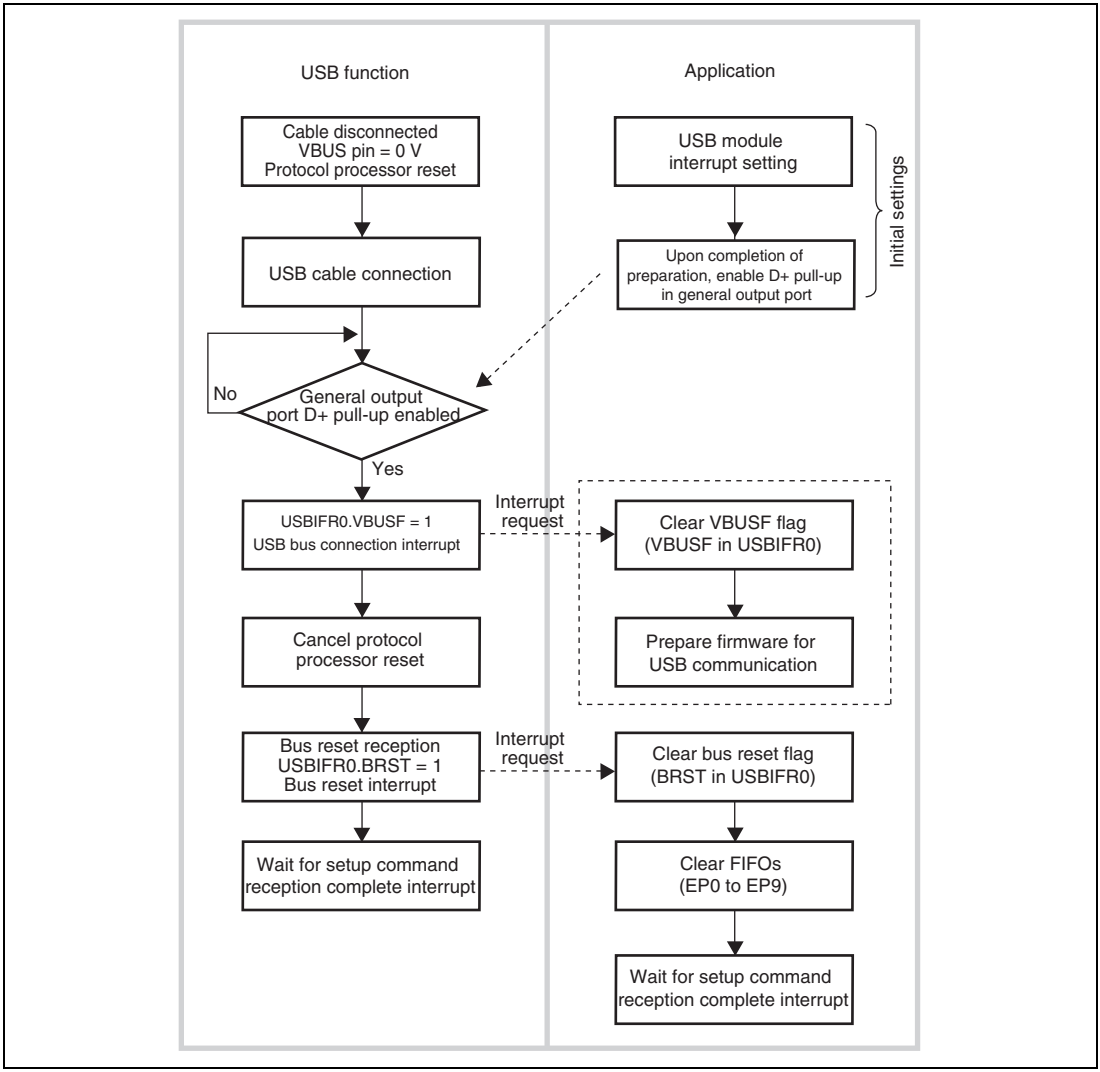


Figure 24.3 Cable Connection Operation

The flowchart in figure 24.3 shows the operation in the case for section 24.10, Example of USB External Circuitry.

In applications that do not require USB cable connection to be detected, processing by the USB bus connection interrupt is not necessary. Preparations should be made with the bus reset interrupt.

24.5.3 Cable Disconnection

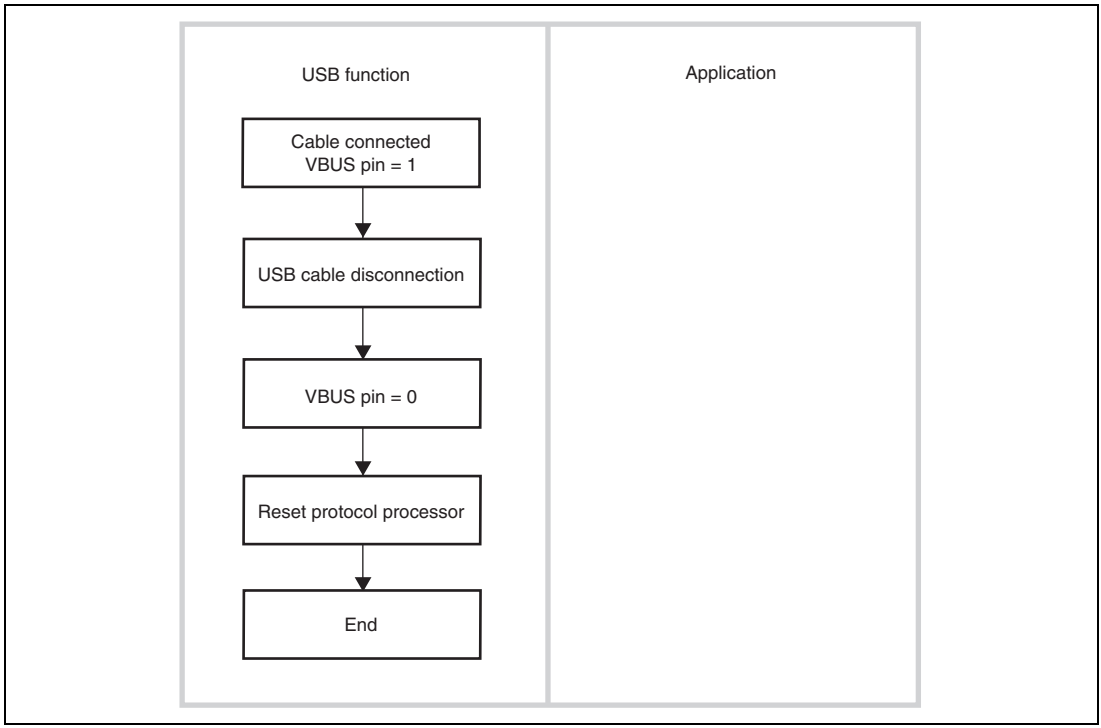


Figure 24.4 Cable Disconnection Operation

The flowchart in figure 24.4 shows the operation in the case for section 24.10, Example of USB External Circuitry.

24.5.4 Control Transfer

Control transfer consists of three stages: setup, data (not always included), and status as illustrated in figure 24.5. The data stage comprises a number of bus transactions. Operation flowcharts for each stage are shown below.

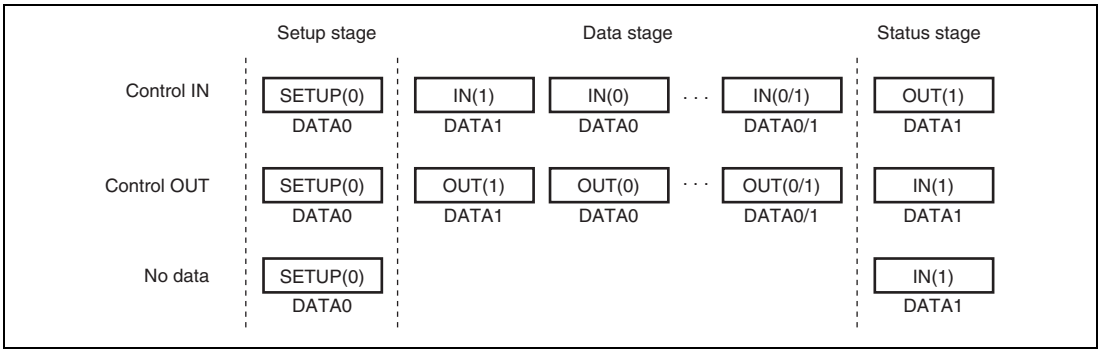
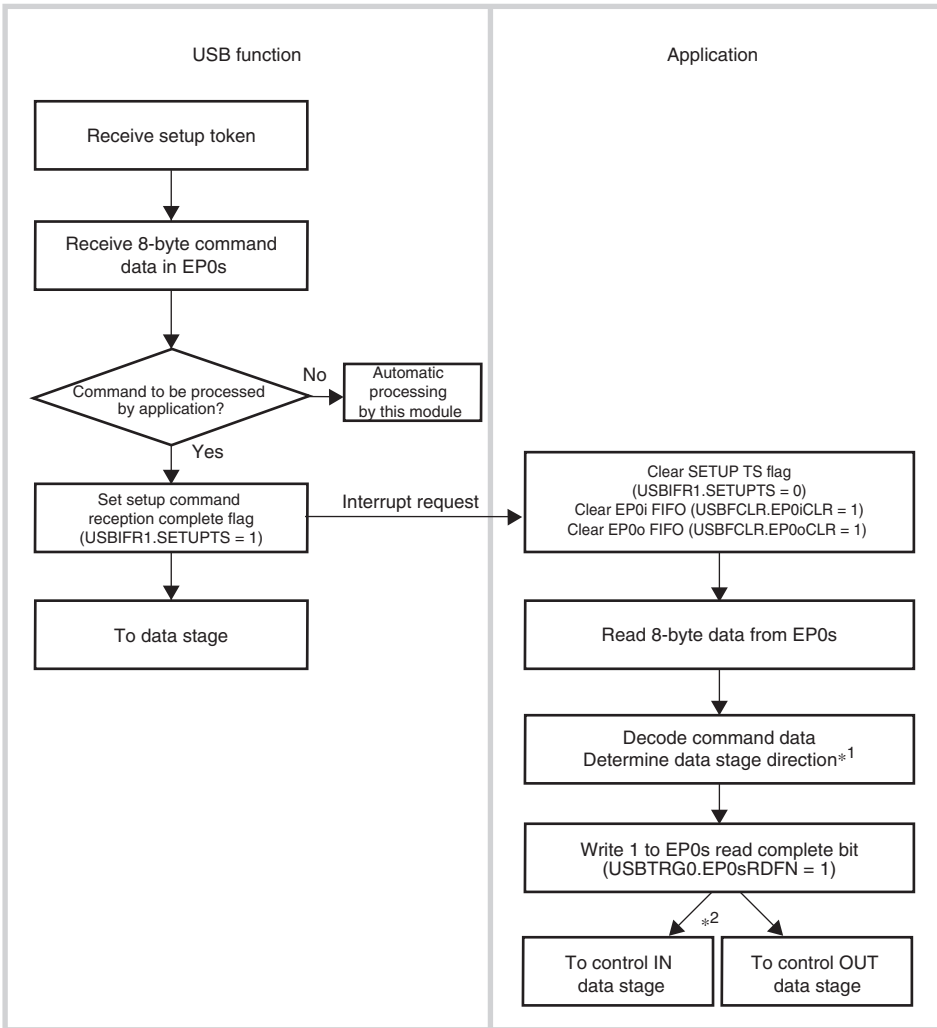


Figure 24.5 Transfer Stages in Control Transfer

(1) Setup Stage



Notes: 1. In the setup stage, the application analyzes command data from the host that must be processed by the application, and determines the subsequent processing method (such as data stage direction).
2. In the case of control OUT transfer direction, enable here EP0i transfer request interrupt required in the status stage. In the case of control IN transfer direction, disable this interrupt as it is not used.

Figure 24.6 Setup Stage Operation

(2) Data Stage (Control-IN)

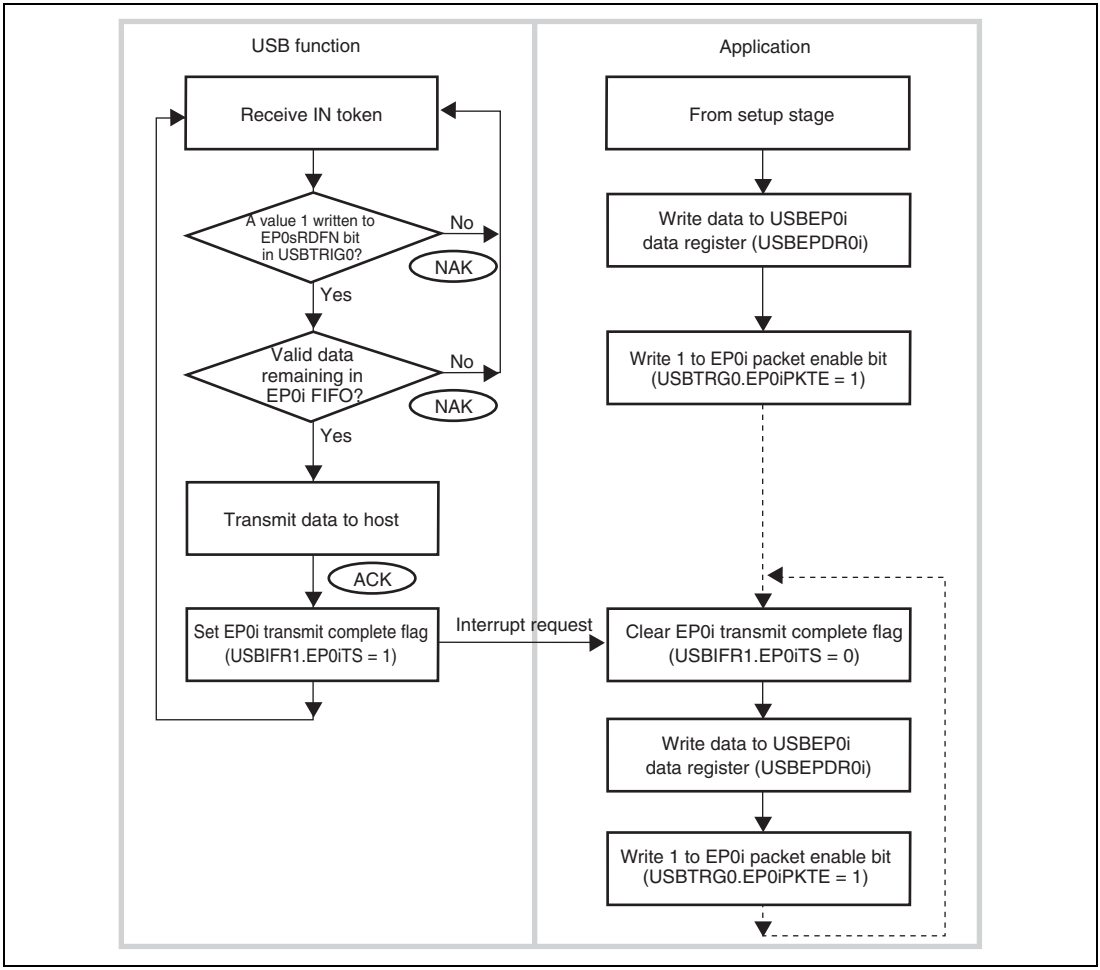


Figure 24.7 Data Stage (Control-IN) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is in-transfer, one packet of data to be sent to the host is written to the FIFO. If there is more data to be sent, this data is written to the FIFO after the data written first has been sent to the host (USBIFR1.EP0iTS = 1).

The end of the data stage is identified when the host transmits an OUT token and the status stage is entered.

Note: If the size of the data transmitted by the function is smaller than the data size requested by the host, the function indicates the end of the data stage by returning to the host a packet shorter than the maximum packet size. If the size of the data transmitted by the function is an integral multiple of the maximum packet size, the function indicates the end of the data stage by transmitting a zero-length packet.

(3) Data Stage (Control-OUT)

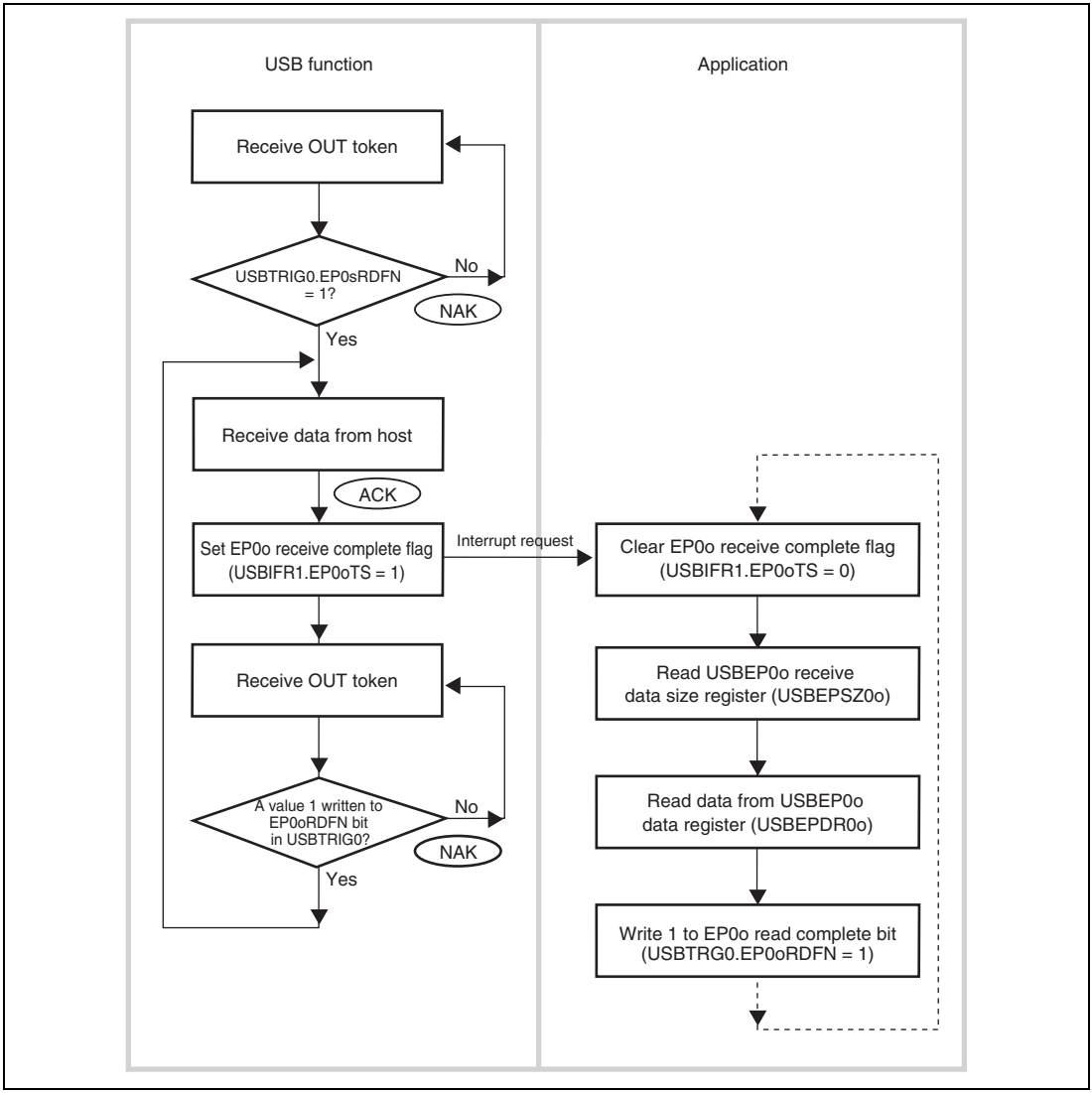


Figure 24.8 Data Stage (Control-OUT) Operation

The application first analyzes command data from the host in the setup stage, and determines the subsequent data stage direction. If the result of command data analysis is that the data stage is OUT-transfer, the application waits for data from the host, and reads data from the FIFO after data is received (USBIFR1.EP0oTS = 1), Then the application writes 1 to the EP0o read complete bit, empties the receive FIFO, and waits for reception of the next data.

The end of the data stage is identified when the host transmits an IN token and the status stage is entered.

(4) Status Stage (Control-IN)

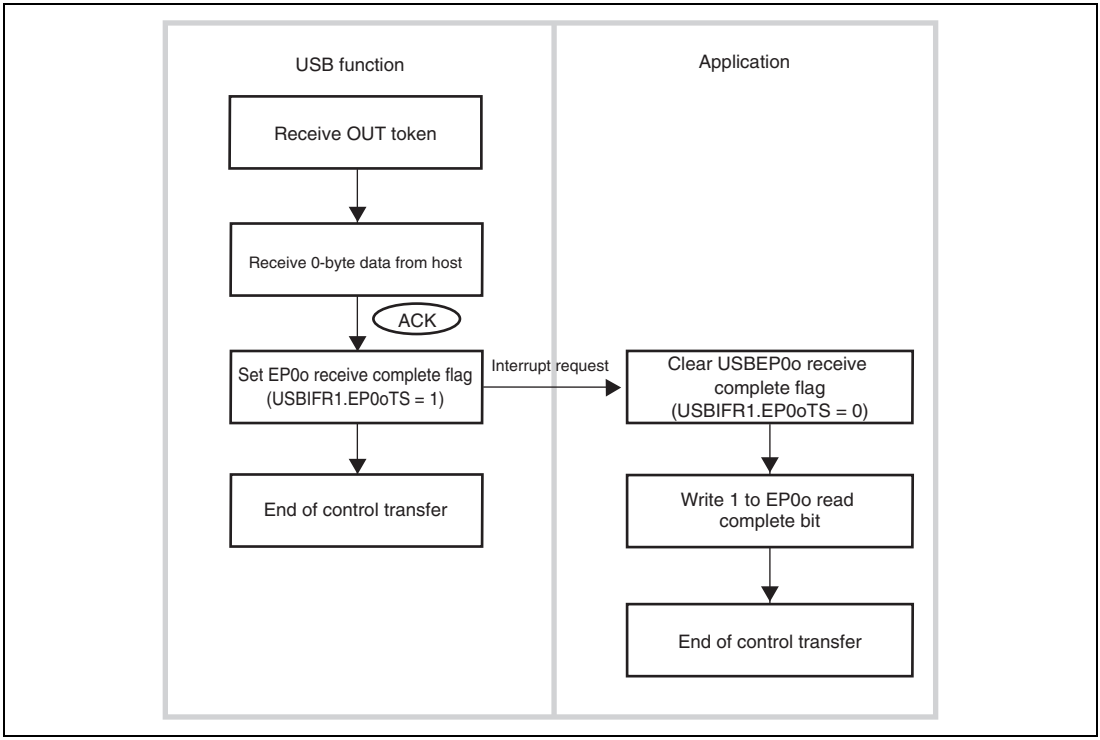


Figure 24.9 Status Stage (Control-IN) Operation

The control-IN status stage starts with an OUT token from the host. The application receives 0-byte data from the host, and ends control transfer.

(5) Status Stage (Control-OUT)

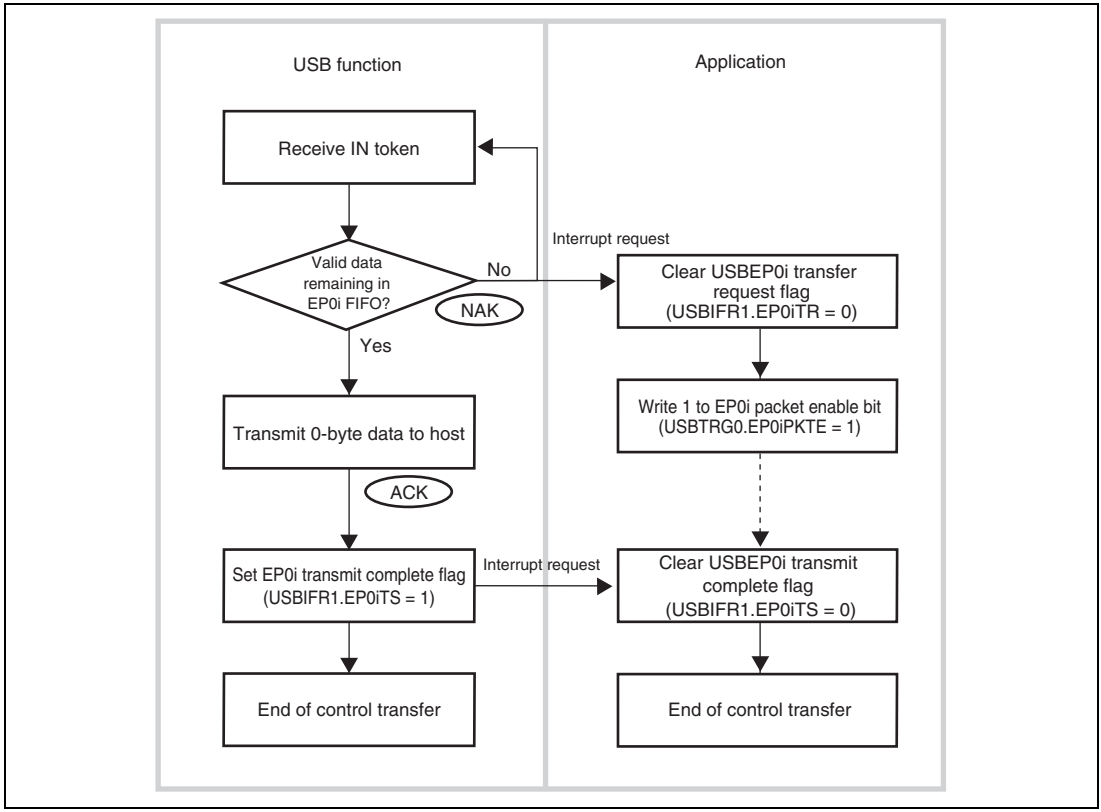


Figure 24.10 Status Stage (Control-OUT) Operation

The control-OUT status stage starts with an IN token from the host. When an IN token is received at the start of the status stage, there is not yet any data in the EP0i FIFO, and so an EP0i transfer request interrupt is generated. The application recognizes from this interrupt that the status stage has started. Next, in order to transmit 0-byte data to the host, 1 is written to the EP0i packet enable bit but no data is written to the EP0i FIFO. As a result, the next IN token causes 0-byte data to be transmitted to the host, and control transfer ends.

After the application has finished all processing relating to the data stage, 1 should be written to the EP0i packet enable bit.

24.5.5 EP1/EP4/EP7 Bulk-OUT Transfer

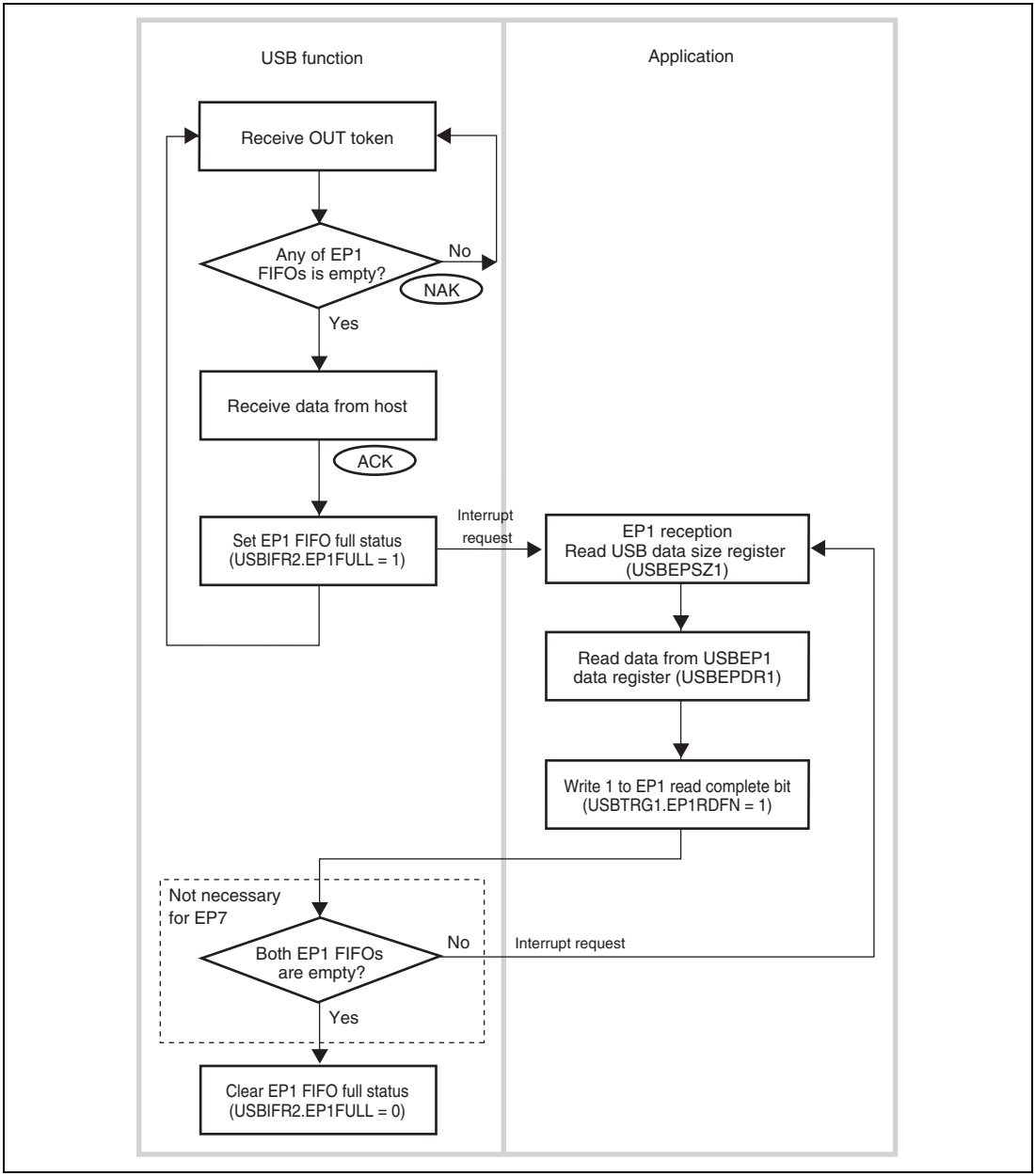


Figure 24.11 EP1 Bulk-OUT Transfer Operation

- Dual FIFOs (EP1, EP4)

  EP1 (EP4) has two 64-byte FIFO buffers, but the user can receive data and read receive data without being aware of this dual-FIFO configuration.

  When one FIFO is full after reception is completed, the EP1 (EP4) FULL bit in USBIFR2 (USBIFR3) is set to 1. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty and so the next packet can be received immediately. When both FIFOs are full, NAK is returned automatically to the host. When reading of the receive data is completed following data reception, 1 is written to the EP1 (EP4) RDFN bit in USBTRG1 (USBTRG2). This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

- Single FIFO (EP7)

  EP7 has a single 64-byte FIFO buffer.

  When the FIFO has received data, the EP7FULL bit in USBIFR4 is set to 1. When the FIFO is full, NAK is returned automatically to the host. When reading of the receive data is completed following data reception, 1 is written to the EP7RDFN bit in USBTRG3. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

24.5.6 EP2/EP5/EP8 Bulk-IN Transfer

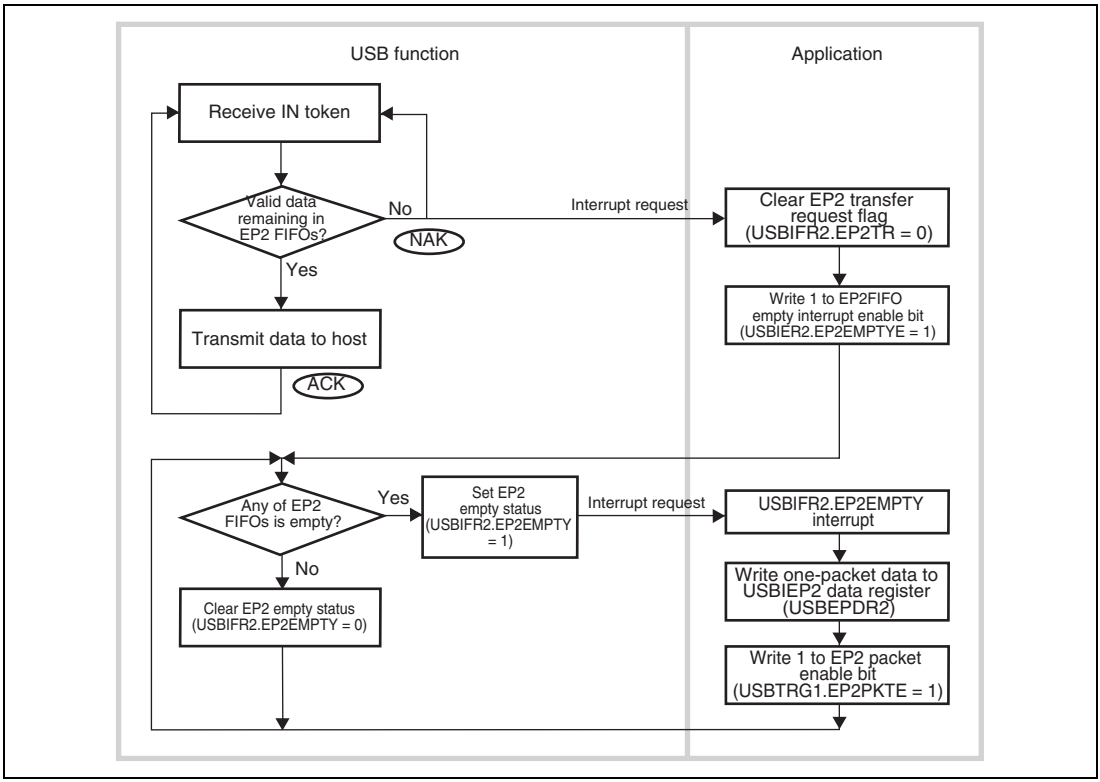


Figure 24.12 EP2 Bulk-IN Transfer Operation

- Dual FIFOs (EP2, EP5)

  EP2 (EP5) has two 64-byte FIFO buffers, but the user can transmit data and write transmit data without being aware of this dual-FIFO configuration. However, one data write should be performed for one FIFO. For example, even if both FIFOs are empty, it is not possible to set the EP2 (EP5) PKTE bit to 1 at one time after consecutively writing 128 bytes of data. The EP2 (EP5) PKTE bit must be set for each 64-byte write.

  When performing bulk-IN transfer, as there is no valid data in the FIFOs on reception of the first IN token, an EP2(EP5)TR interrupt in USBIFR2 (USBIFR3) is requested. With this interrupt, 1 is written to the EP2 (EP5) EMPTYE bit in USBIER2 (USBIER3), and the EP2 (EP5) FIFO empty interrupt is enabled. At first, both EP2 (EP5) FIFOs are empty, and so an EP2 (EP5) FIFO empty interrupt is generated immediately.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write for one FIFO, the other FIFO is empty and so the next transmit data can be written immediately to the other FIFO. When both FIFOs are full, EP2 (EP5) EMPTYE is cleared to 0. If at least one FIFO is empty, the EP2 (EP5) EMPTY bit in USBIFR2 (USBIFR3) is set to 1. When ACK is returned from the host after data transmission is completed, the FIFO that has transmitted data becomes empty. If the other FIFO contains valid transmit data at this time, transmission can be continued.

When transmission of all data has been completed, write 0 to the EP2 (EP5) EMPTYE bit in USBIER2 (USBIER3) to disable interrupt requests.

- Single FIFO (EP8)

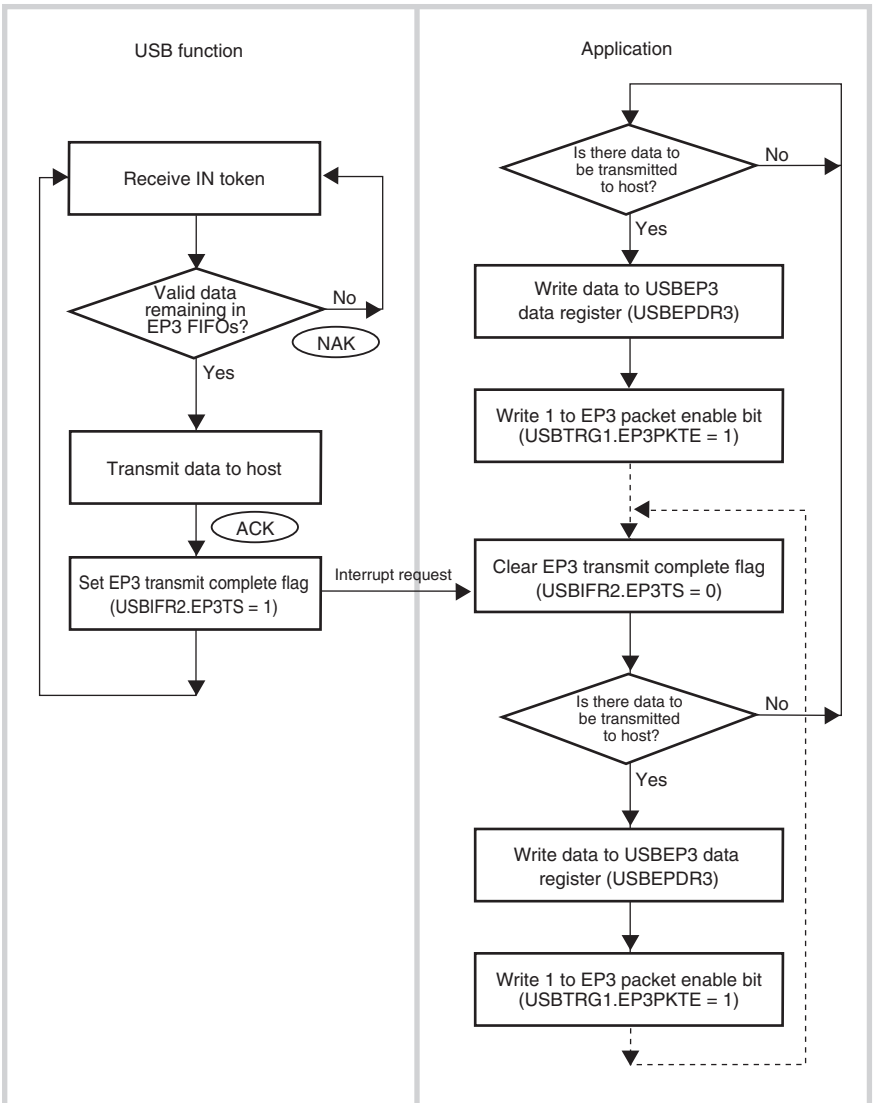
  EP8 has a single 64-byte FIFO buffer.

  When performing bulk-IN transfer, as there is no valid data in the FIFO on reception of the first IN token, an EP8TR interrupt in USBIFR4 is requested. With this interrupt, 1 is written to the EP8EMPTYE bit in USBIER4, and the EP8 FIFO empty interrupt is enabled.

  The data to be transmitted is written to the data register using this interrupt. When the FIFO is full, EP8EMPTYE is cleared to 0. When ACK is returned from the host after data transmission is completed, the FIFO that has transmitted data becomes empty and the EP8EMPTY bit in USBIFR4 is set to 1.

  When transmission of all data has been completed, write 0 to the EP8EMPTYE bit in USBIER4 to disable interrupt requests.

24.5.7 EP3/EP6/EP9 Interrupt-IN Transfer



Note: This flowchart shows an example of interrupt transfer processing. However, another flow can be considered: when there is data to be transmitted, FIFO emptiness is checked by reading the EP3DE bit in the USB data status register and then data is written to the FIFO.

Figure 24.13 EP3 Interrupt-IN Transfer Operation

24.6 Processing of USB Standard Commands and Class/Vendor Commands

24.6.1 Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing on the application side. Commands that require or do not require decoding on the application side are listed in table 24.8 below.

Table 24.8 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear Feature	Get Descriptor
Get Configuration	Class/Vendor commands
Get Interface	Set Descriptor
Get Status	Sync Frame
Set Address	
Set Configuration	
Set Feature	
Set Interface	

If decoding is not necessary on the application side, command decoding, data stage processing, and status stage processing are performed automatically. Therefore no processing is necessary for the user, and no interrupt is generated in this case.

If decoding is necessary on the application side, the USB function module stores the command in the EP0s FIFO. After normal reception is completed, the SETUPTS flag in USBIFR1 is set to 1 and an interrupt request is generated. In this interrupt routine, 8-byte data must be read from the USBEP0s data register (USBEPDR0s) and decoded by the firmware program. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

24.7 Stall Operations

24.7.1 Overview

This section describes stall operations in the USB function module. The USB function module stall function is used in the following cases:

- When the application forcibly stalls an endpoint for some reason
- When a stall is performed automatically within the USB function module due to a USB specification violation

The USB function module has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application. They must be cleared with a Clear Feature command from the host. The internal status bit for EP0 is automatically cleared only when the setup command is received.

24.7.2 Forcible Stall by Application

The application uses the USBEPSTL register to issue a stall request for the USB function module. When the application wishes to stall a specific endpoint, it sets the corresponding bit in USBEPSTL (1-1 in figure 24.14). The internal status bits remain unchanged at this time. When a transaction is sent from the host to the endpoint for which the USBEPSTL bit is set, the USB function module references the internal status bit, and if this is not set, references the corresponding bit in USBEPSTL (1-2 in figure 24.14). If the corresponding bit in USBEPSTL is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 24.14). If the corresponding bit in USBEPSTL is not set, the internal status bit remains unchanged and the transaction is accepted.

Once an internal status bit is set, it remains set until it is cleared by a Clear Feature command from the host, without regard to the USBEPSTL register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 24.14), the USB function module continues to return a stall handshake while the bit in USBEPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 24.14). To clear a stall, therefore, the corresponding bit in USBEPSTL must be cleared by the application, and the internal status bit must be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 24.14).

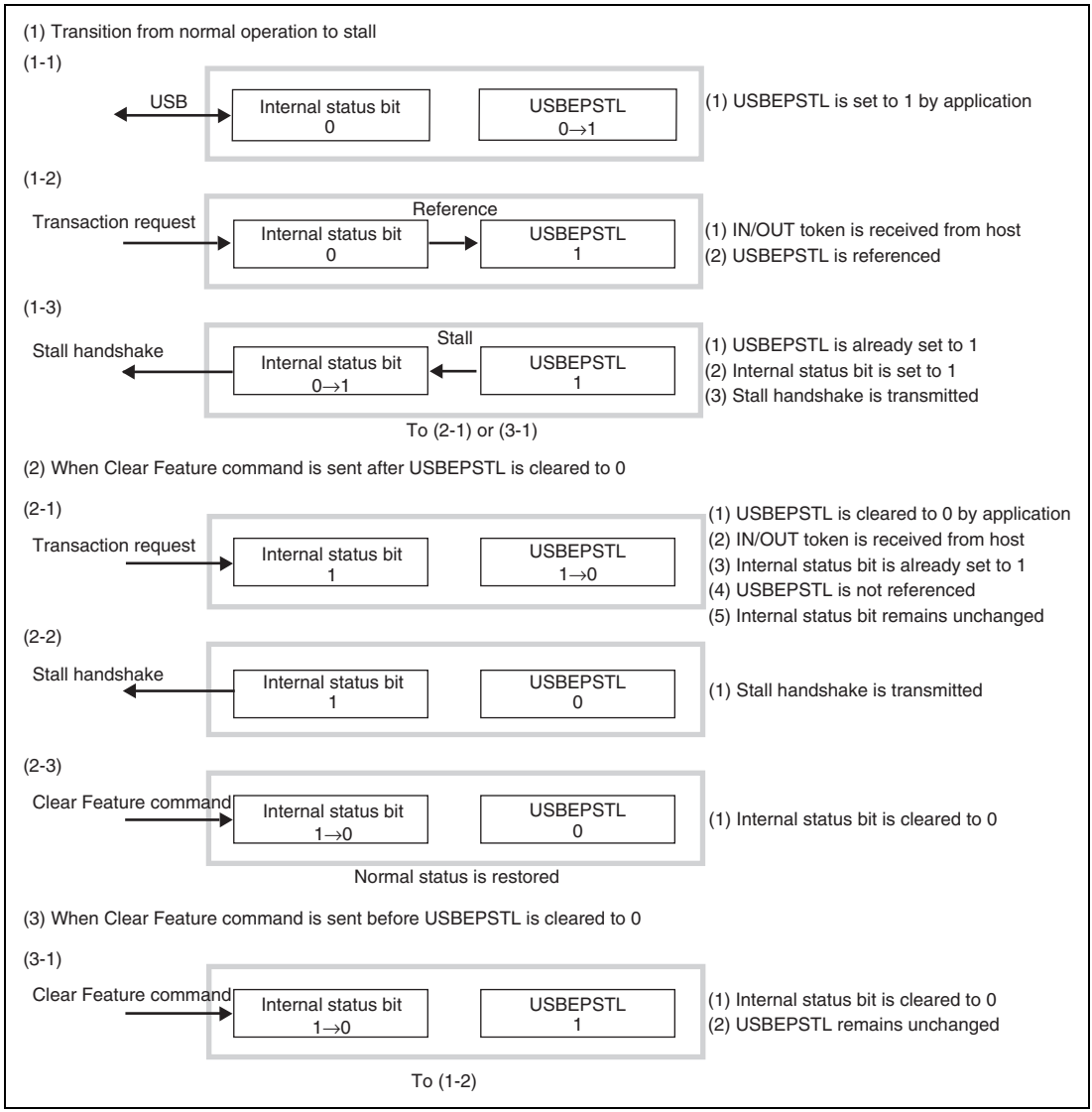


Figure 24.14 Forcible Stall by Application

24.7.3 Automatic Stall by USB Function Module

When a stall setting is made with a Set Feature command, or in the event of a USB specification violation, the USB function module automatically sets the internal status bit for the relevant endpoint regardless of the USBEPSTL register setting, and returns a stall handshake (1-1 in figure 24.15).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host regardless of the USBEPSTL register setting. After a bit is cleared by the Clear Feature command, USBEPSTL is referenced (3-1 in figure 24.15). The USB function module continues to return a stall handshake while the internal status bit is set to 1, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 24.15). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 24.15). If set by the application, USBEPSTL should also be cleared (2-1 in figure 24.15).

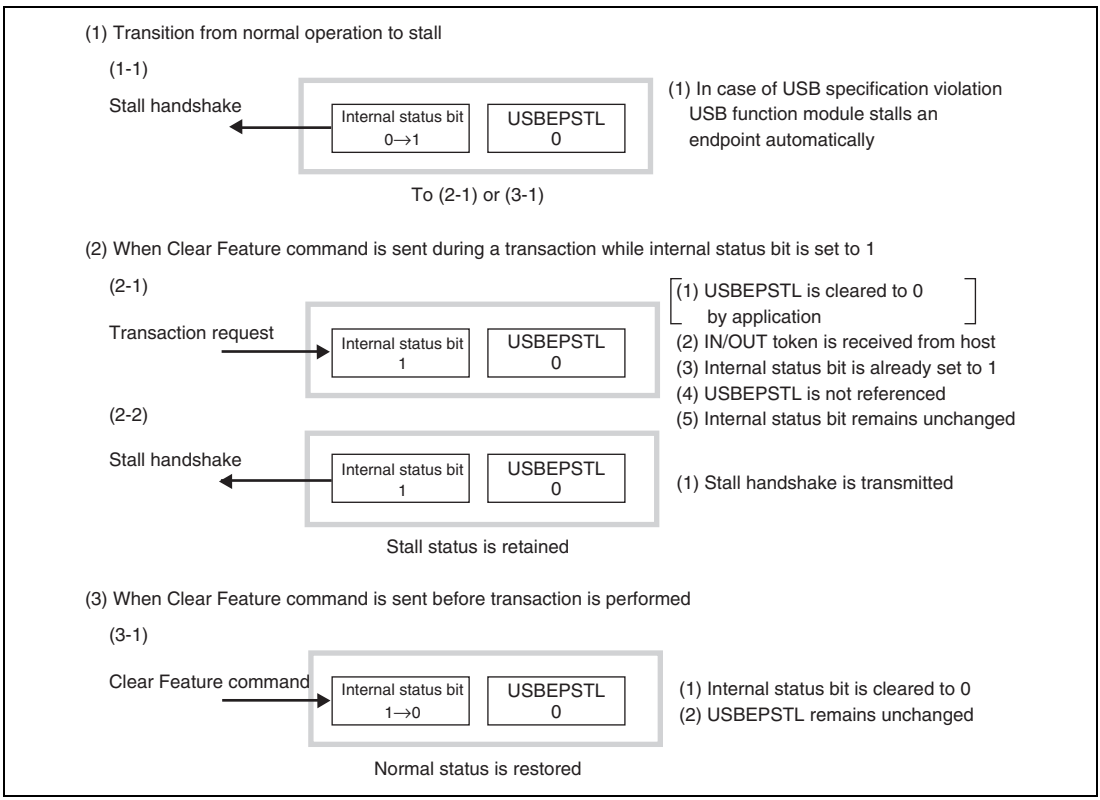


Figure 24.15 Automatic Stall by USB Function Module

24.8 DMA Transfer

24.8.1 Overview

This module allows DMA transfer for endpoints 1, 2, 4, and 5, excluding transfer of word and longword. If endpoint 1 contains at least one byte of valid receive data, a DMA transfer request is issued to endpoint 1. If there is no valid data in endpoint 2, a DMA transfer request is issued to endpoint 2. If endpoint 4 contains at least one byte of valid receive data, a DMA transfer request is issued to endpoint 4. If there is no valid data in endpoint 5, a DMA transfer request is issued to endpoint 5.

When EP1DMAE or EP4DMAE in the USBDMA setting register is set to 1 to allow DMA transfer, 0-length data received for endpoint 1 or 4 is ignored. When DMA transfer is set, it is unnecessary to write 1 to the EP1RDFN, EP2PKTE, EP4RDFN, and EP5PKTE bits in USBTRG1 or USBTRG2. (However, the PKTE bit in USBTRG1 or USBTRG2 must be set to 1 for data with a size less than the maximum number of bytes.) For EP1 and EP4, the FIFO buffer automatically becomes empty when the received data has been completely read. For EP2 and EP5, the FIFO automatically becomes full when the maximum number of bytes (64 bytes) is written to the FIFO, allowing the data in the FIFO to be transmitted. (See figures 24.16 and 24.19.)

24.8.2 DMA Transfer for Endpoints 1 and 4

If the received data for EP1 is transferred by DMA, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed in the module. Therefore, do not write 1 to the EP1RDFN bit in USBTRG1 after reading the data on one side of the FIFO. If 1 is written to the EP1RDFN bit, correct operation cannot be guaranteed.

For example, if 150-byte data is received from the host, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed internally in the three places in figure 24.16. Since this processing is performed when the data on the currently selected FIFO becomes empty, the processing is automatically performed in the same way even if data of 64 bytes or less is transferred.

Similarly, if the received data for EP4 is transferred by DMA, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP4RDFN bit in USBTRG2 is automatically performed in the module. Therefore, do not write 1 to the EP4RDFN bit in USBTRG2 after reading the data on one side of the FIFO. If 1 is written to the EP4RDFN bit, correct operation cannot be guaranteed.

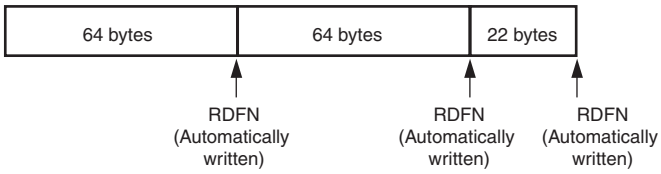


Figure 24.16 EP1/EP4 RDFN (EP1RDFN, EP4RDFN) Operation

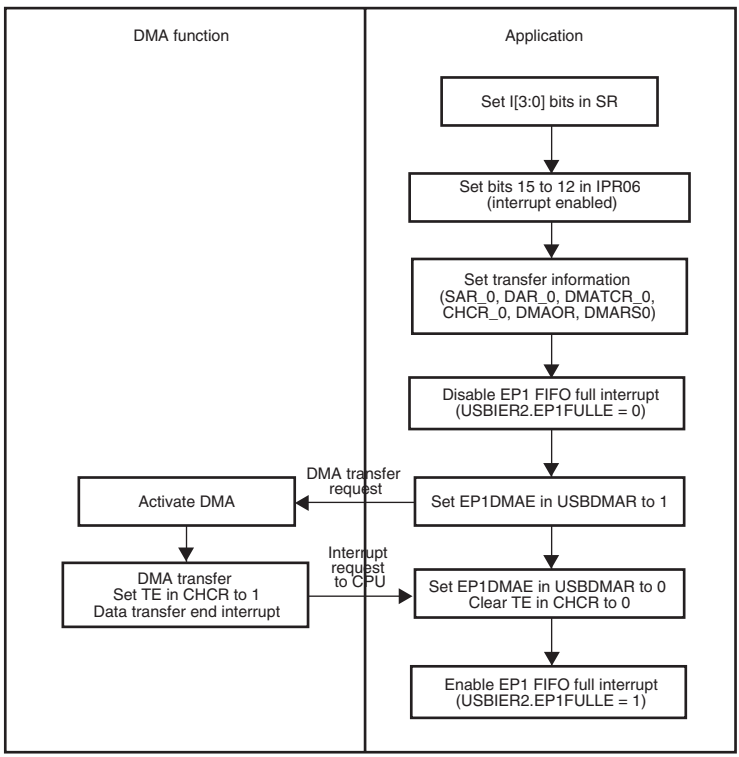
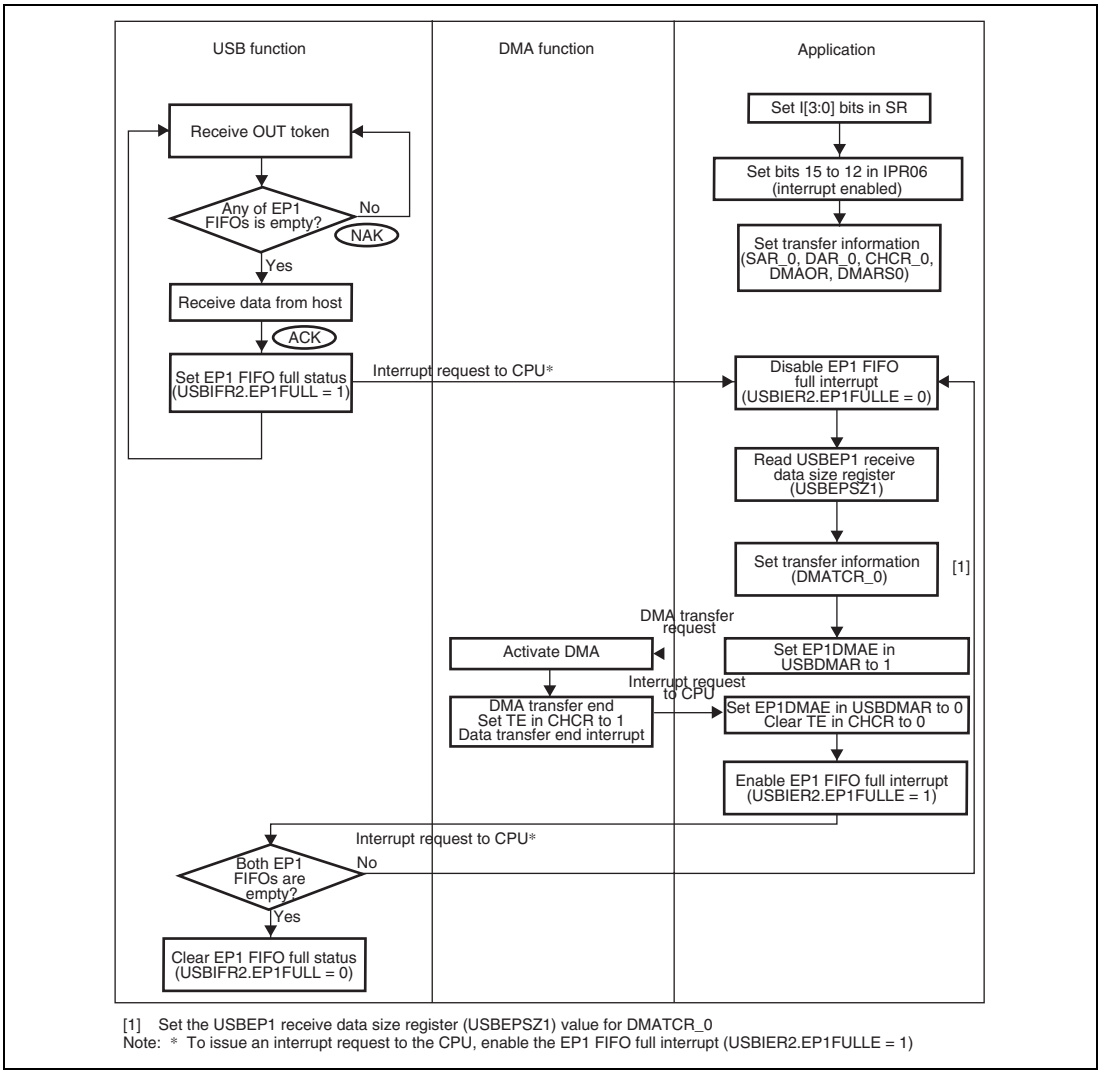


**Figure 24.17 Example of DMA Transfer (Channel 0) for Bulk-OUT Transfer (EP1)
(When Receive Data Size is Determined Before Receiving OUT Token)**



[1] Set the USBEP1 receive data size register (USBEPSZ1) value for DMATCR_0
Note: * To issue an interrupt request to the CPU, enable the EP1 FIFO full interrupt (USBIER2.EP1FULLE = 1)

Figure 24.18 Example of DMA Transfer (Channel 0) for Bulk-OUT Transfer (EP1) (When Receive Data Size Cannot be Determined Before Receiving OUT Token)

24.8.3 DMA Transfer for Endpoints 2 and 5

If the transmitted data for EP2 is transferred by DMA, when the data on one side of FIFO (64 bytes) becomes full, processing equivalent to writing 1 to the EP2PKTE bit in USBTRG1 is automatically performed in the module. Therefore, when data to be transferred is a multiple of 64 bytes, writing 1 to the EP2PKTE bit in USBTRG1 is not necessary.

For data less than 64 bytes, a 1 should be written to the EP2PKTE bit in USBTRG1 by a DMA transfer end interrupt of the DMAC. If a 1 is written to the EP2PKTE bit for transferring the maximum number of bytes (64 bytes), the correct operation cannot be guaranteed.

For example, if 150-byte data is transmitted to the host, processing equivalent to writing 1 to the EP2PKTE bit in USBTRG1 is automatically performed internally in the two places in figure 24.19. Since this processing is performed when the data on the currently selected FIFO becomes full, the processing is automatically performed only when 64-byte data is transferred.

When the last 22 bytes have been transferred, write 1 to the EP2PKTE bit in USBTRG1 by the software because this is not automatically executed. There is no data to be transferred on the application side, but this module outputs the DMA transfer request for EP2 as long as the FIFO has a space. When the data has completely been transferred by DMA, write 0 to the EP2DMAE bit in USBDMAR to cancel the DMA transfer request for EP2.

Similarly, if the transmitted data for EP5 is transferred by DMA, when the data on one side of FIFO (64 bytes) becomes full, processing equivalent to writing 1 to the EP5PKTE bit in USBTRG2 is automatically performed in the module. Therefore, when data to be transferred is a multiple of 64 bytes, writing 1 to the EP5PKTE bit in USBTRG2 is not necessary. For data less than 64 bytes, a 1 should be written to the EP5PKTE bit in USBTRG2 by a DMA transfer end interrupt of the DMAC. If a 1 is written to the EP5PKTE bit for transferring the maximum number of bytes (64 bytes), the correct operation cannot be guaranteed.

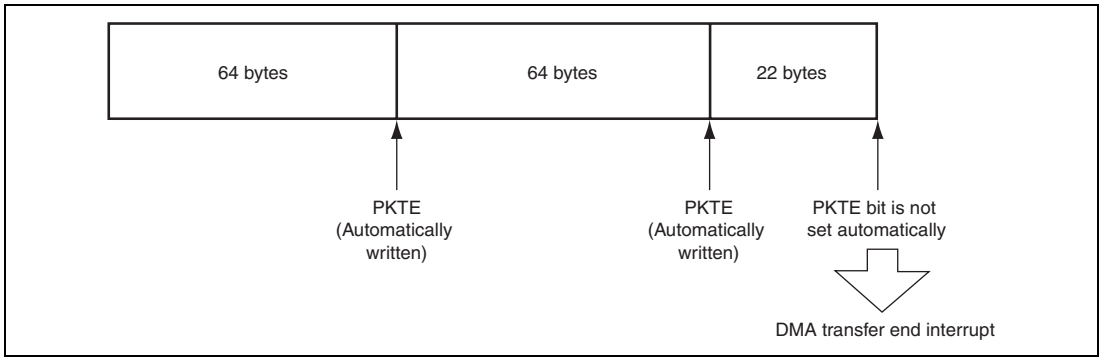
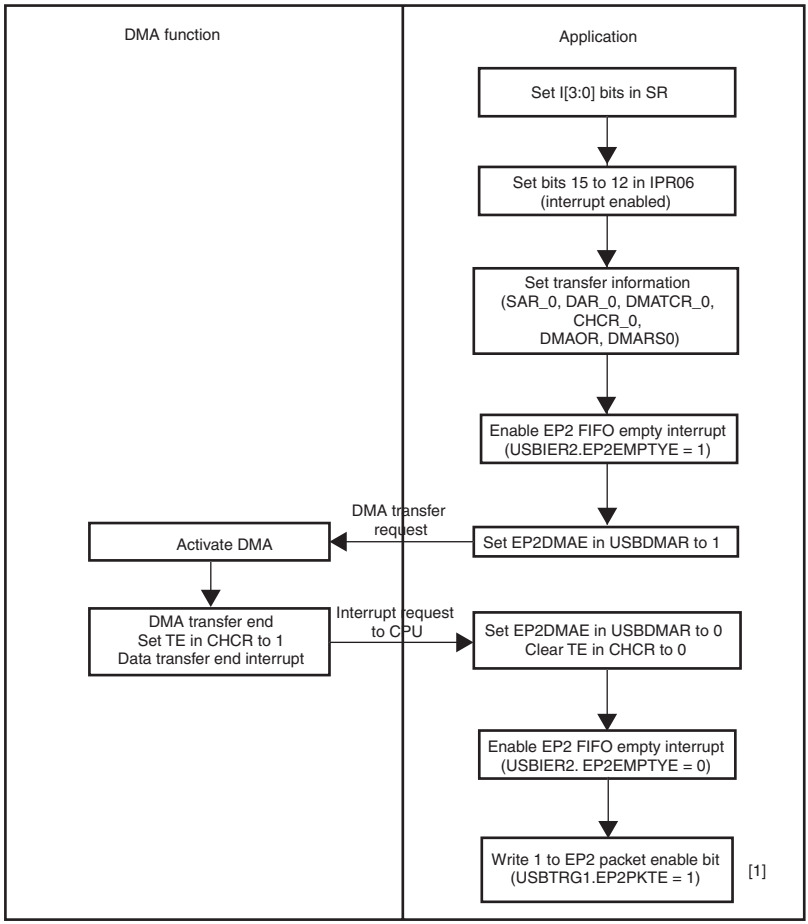
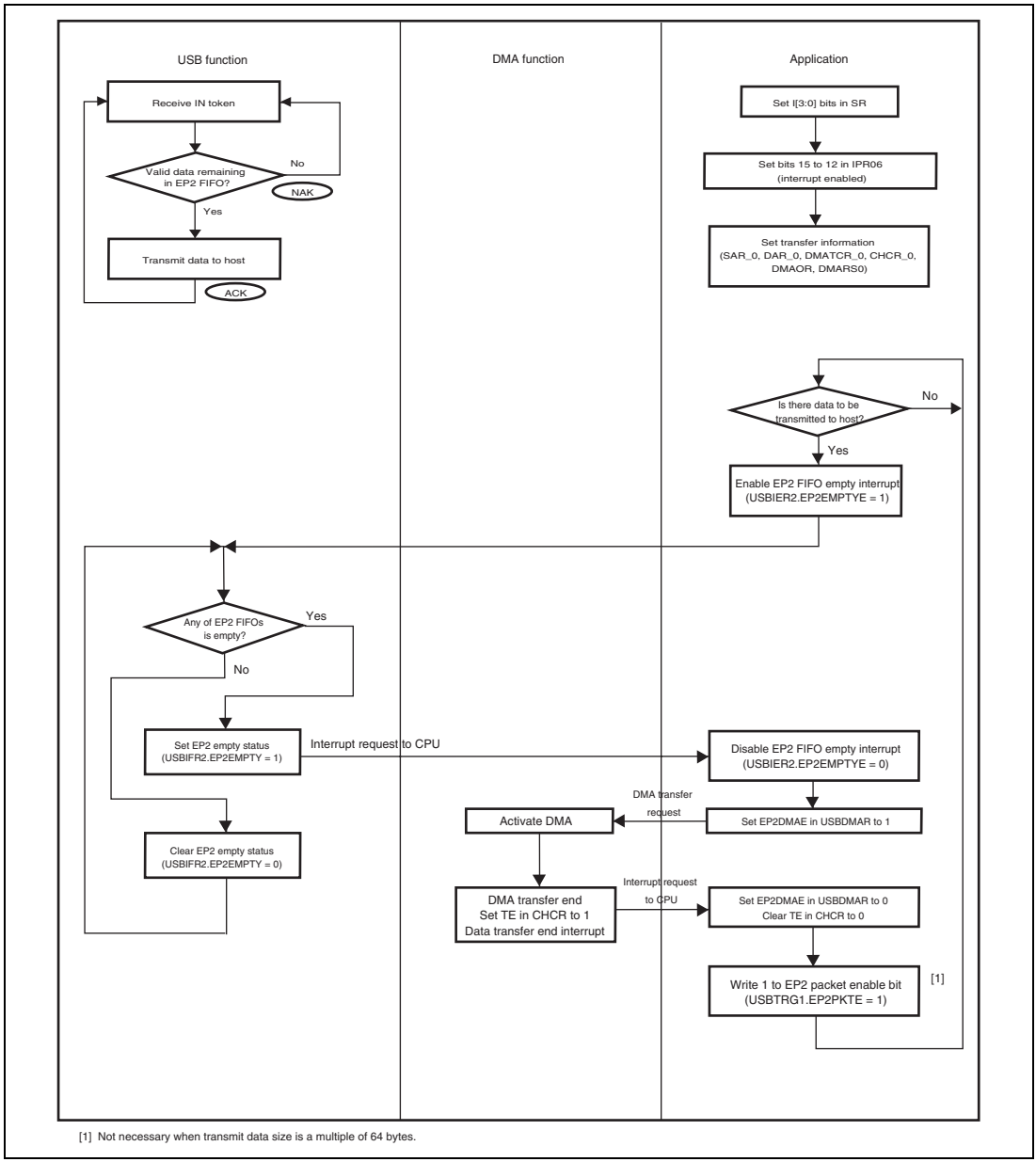


Figure 24.19 EP2/EP5 PKTE (EP2PKTE, EP5PKTE) Operation



[1] Not necessary when transmit data size is a multiple of 64 bytes.

Figure 24.20 Example of DMA Transfer (Channel 0) for Bulk-IN Transfer (EP2) (When Transmit Data Size is Determined Before Receiving IN Token)



[1] Not necessary when transmit data size is a multiple of 64 bytes.

Figure 24.21 Example of DMA Transfer (Channel 0) for Bulk-IN Transfer (EP2) (When Transmit Data Size Cannot be Determined Before Receiving IN Token)

24.9 DTC Transfer

This module allows DTC transfer for endpoints 1, 2, 4, and 5, excluding transfer of word and longword. If endpoint 1 contains at least one byte of valid receive data, a DTC transfer request is issued to endpoint 1. If there is no valid data in endpoint 2, a DTC transfer request is issued to endpoint 2. If endpoint 4 contains at least one byte of valid receive data, a DTC transfer request is issued to endpoint 4. If there is no valid data in endpoint 5, a DTC transfer request is issued to endpoint 5.

When EP1DMAE or EP4DMAE in the USBDMA setting register is set to 1 to allow DTC transfer, 0-length data received for endpoint 1 or 4 is ignored. When DTC transfer is set, it is unnecessary to write 1 to the EP1RDFN, EP2PKTE, EP4RDFN, and EP5PKTE bits in USBTRG1 or USBTRG2. (However, the PKTE bit in USBTRG1 or USBTRG2 must be set to 1 for data with a size less than the maximum number of bytes.) For EP1 and EP4, the FIFO buffer automatically becomes empty when the received data has been completely read. For EP2 and EP5, the FIFO automatically becomes full when the maximum number of bytes (64 bytes) is written to the FIFO, allowing the data in the FIFO to be transmitted. (See figures 24.22 and 24.25.)

24.9.1 DTC Transfer for Endpoints 1 and 4

If the received data for EP1 is transferred by DTC, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed in the module. Therefore, do not write 1 to the EP1RDFN bit in USBTRG1 after reading the data on one side of the FIFO. If 1 is written to the EP1RDFN bit, correct operation cannot be guaranteed.

For example, if 150-byte data is received from the host, processing equivalent to writing 1 to the EP1RDFN bit in USBTRG1 is automatically performed internally in the three places in figure 24.22. Since this processing is performed when the data on the currently selected FIFO becomes empty, the processing is automatically performed in the same way even if data of 64 bytes or less is transferred.

Similarly, if the received data for EP4 is transferred by DTC, when the currently selected data FIFO becomes empty, processing equivalent to writing 1 to the EP4RDFN bit in USBTRG2 is automatically performed in the module. Therefore, do not write 1 to the EP4RDFN bit in USBTRG2 after reading the data on one side of the FIFO. If 1 is written to the EP4RDFN bit, correct operation cannot be guaranteed.

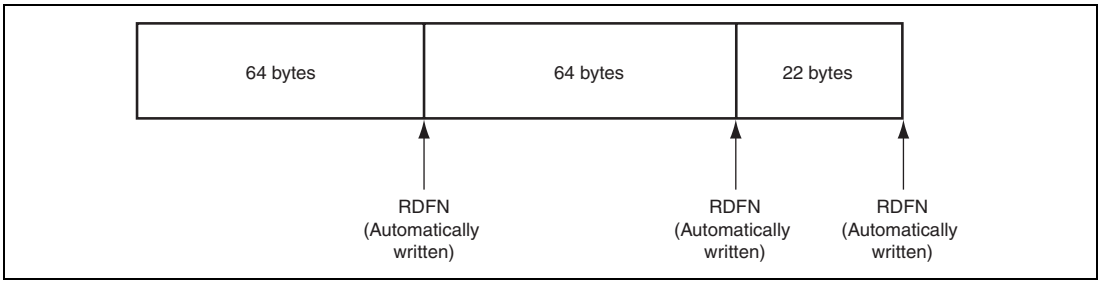
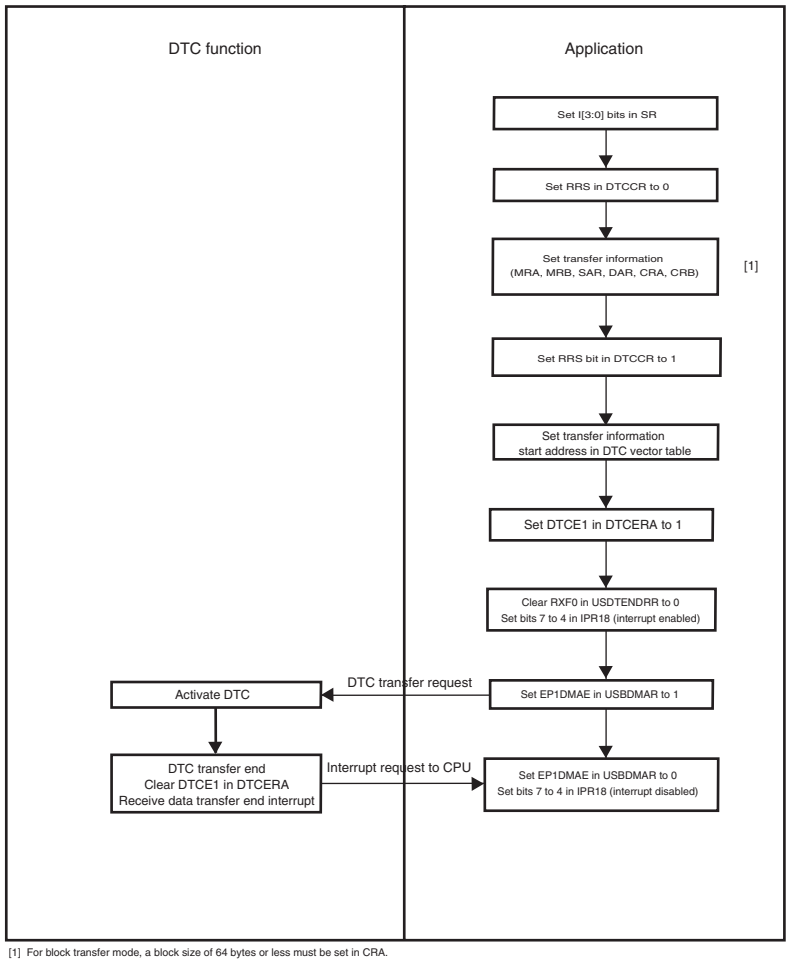


Figure 24.22 EP1/EP4 RDFN (EP1RDFN, EP4RDFN) Operation



[1] For block transfer mode, a block size of 64 bytes or less must be set in CRA.

Figure 24.23 Example of DTC Transfer for Bulk-OUT Transfer (EP1) (When Receive Data Size is Determined Before Receiving OUT Token)

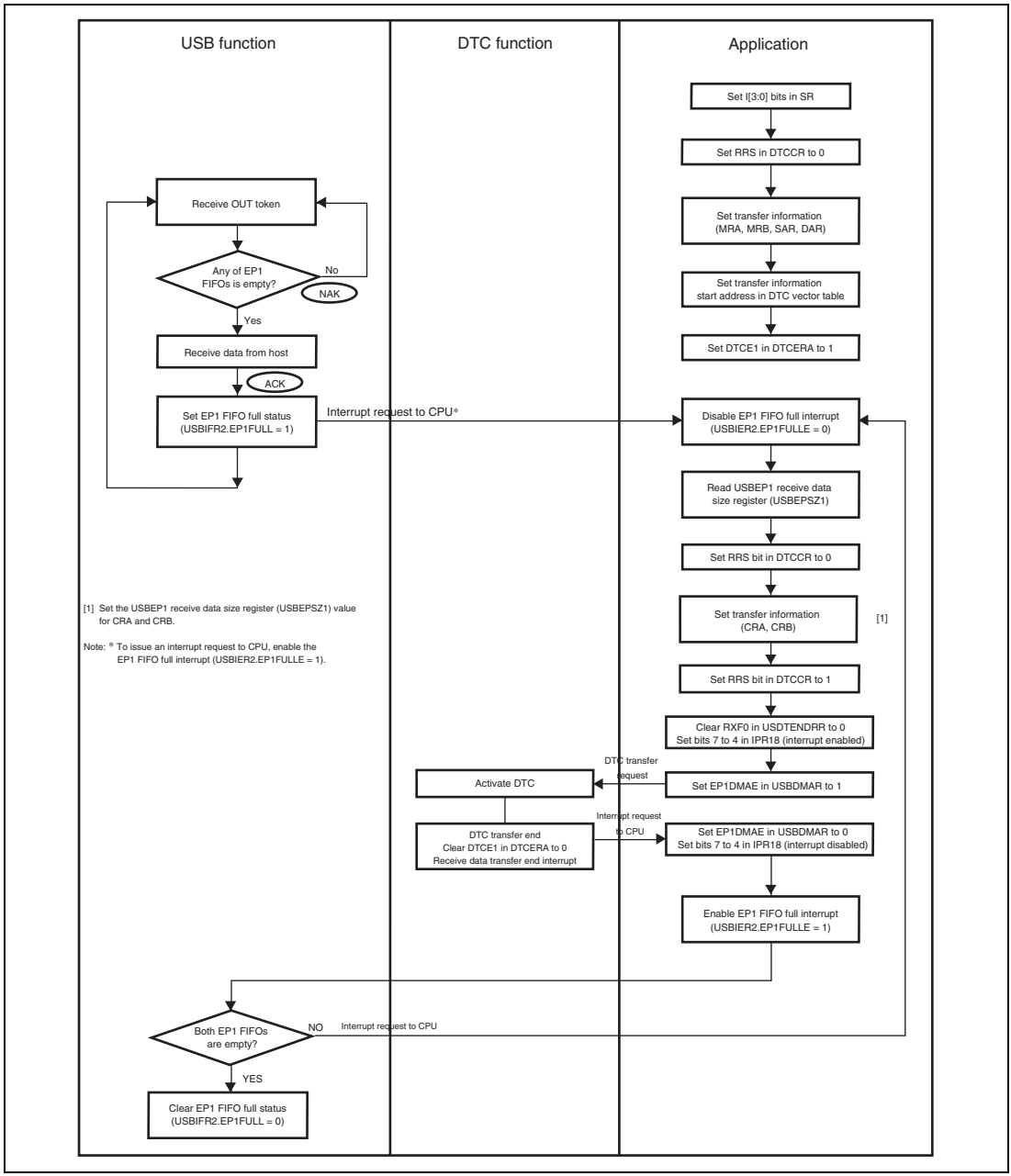


Figure 24.24 Example of DTC Transfer for Bulk-OUT Transfer (EP1) (When Receive Data Size Cannot be Determined Before Receiving OUT Token)

24.9.2 DTC Transfer for Endpoints 2 and 5

If the transmitted data for EP2 is transferred by DTC, when the data on one side of FIFO (64 bytes) becomes full, processing equivalent to writing 1 to the EP2PKTE bit in USBTRG1 is automatically performed in the module. Therefore, when data to be transferred is a multiple of 64 bytes, writing 1 to the EP2PKTE bit in USBTRG1 is not necessary.

For data less than 64 bytes, a 1 should be written to the EP2PKTE bit in USBTRG1 by a DTC transfer end interrupt of the DTC. If a 1 is written to the EP2PKTE bit for transferring the maximum number of bytes (64 bytes), the correct operation cannot be guaranteed.

For example, if 150-byte data is transmitted to the host, processing equivalent to writing 1 to the EP2PKTE bit in USBTRG1 is automatically performed internally in the two places in figure 24.25. Since this processing is performed when the data on the currently selected FIFO becomes full, the processing is automatically performed only when 64-byte data is transferred.

When the last 22 bytes have been transferred, write 1 to the EP2PKTE bit in USBTRG1 by the software because this is not automatically executed. There is no data to be transferred on the application side, but this module outputs the DTC transfer request for EP2 as long as the FIFO has a space. When the data has completely been transferred by DTC, write 0 to the EP2DMAE bit in USBDMAR to cancel the DTC transfer request for EP2.

Similarly, if the transmitted data for EP5 is transferred by DTC, when the data on one side of FIFO (64 bytes) becomes full, processing equivalent to writing 1 to the EP5PKTE bit in USBTRG2 is automatically performed in the module. Therefore, when data to be transferred is a multiple of 64 bytes, writing 1 to the EP5PKTE bit in USBTRG2 is not necessary. For data less than 64 bytes, a 1 should be written to the EP5PKTE bit in USBTRG2 by a DTC transfer end interrupt of the DTC. If a 1 is written to the EP5PKTE bit for transferring the maximum number of bytes (64 bytes), the correct operation cannot be guaranteed.

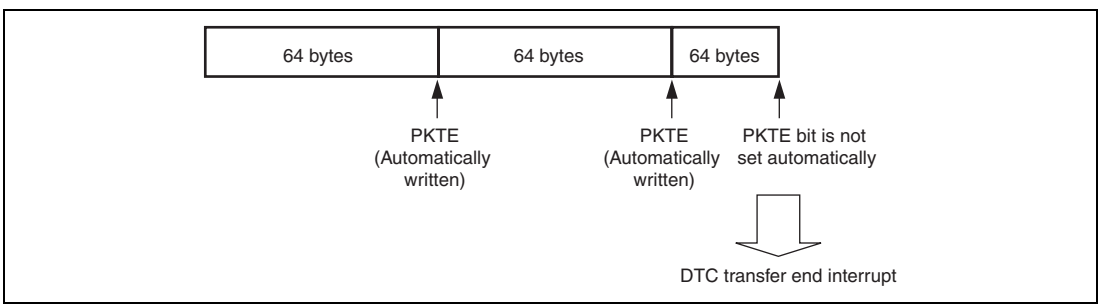
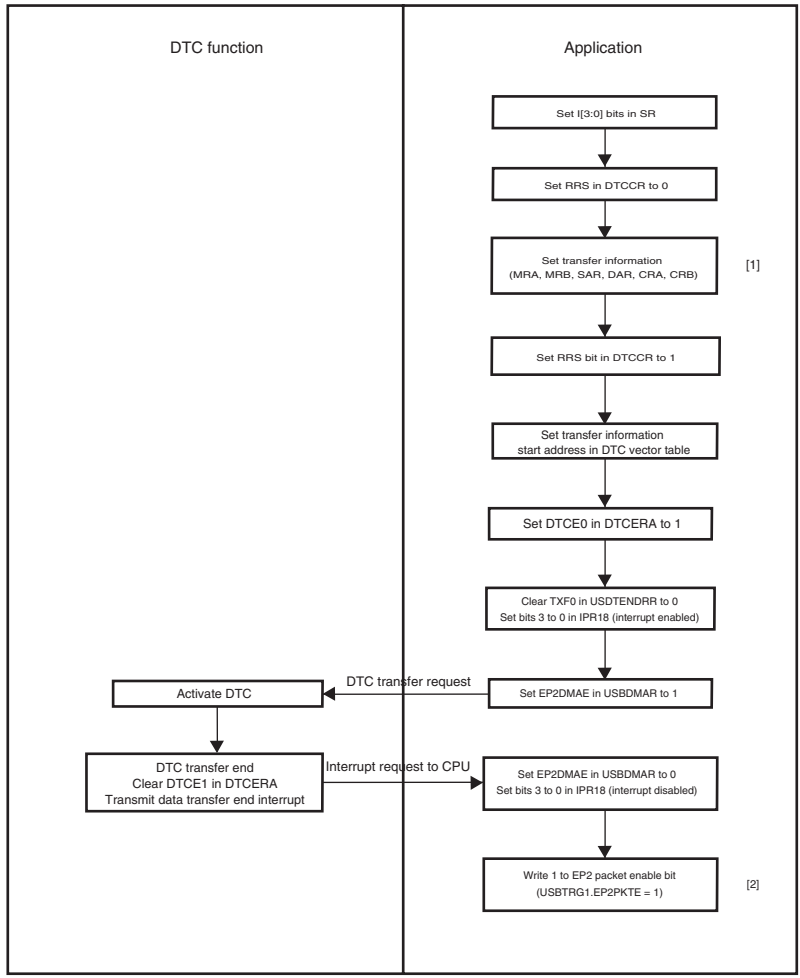


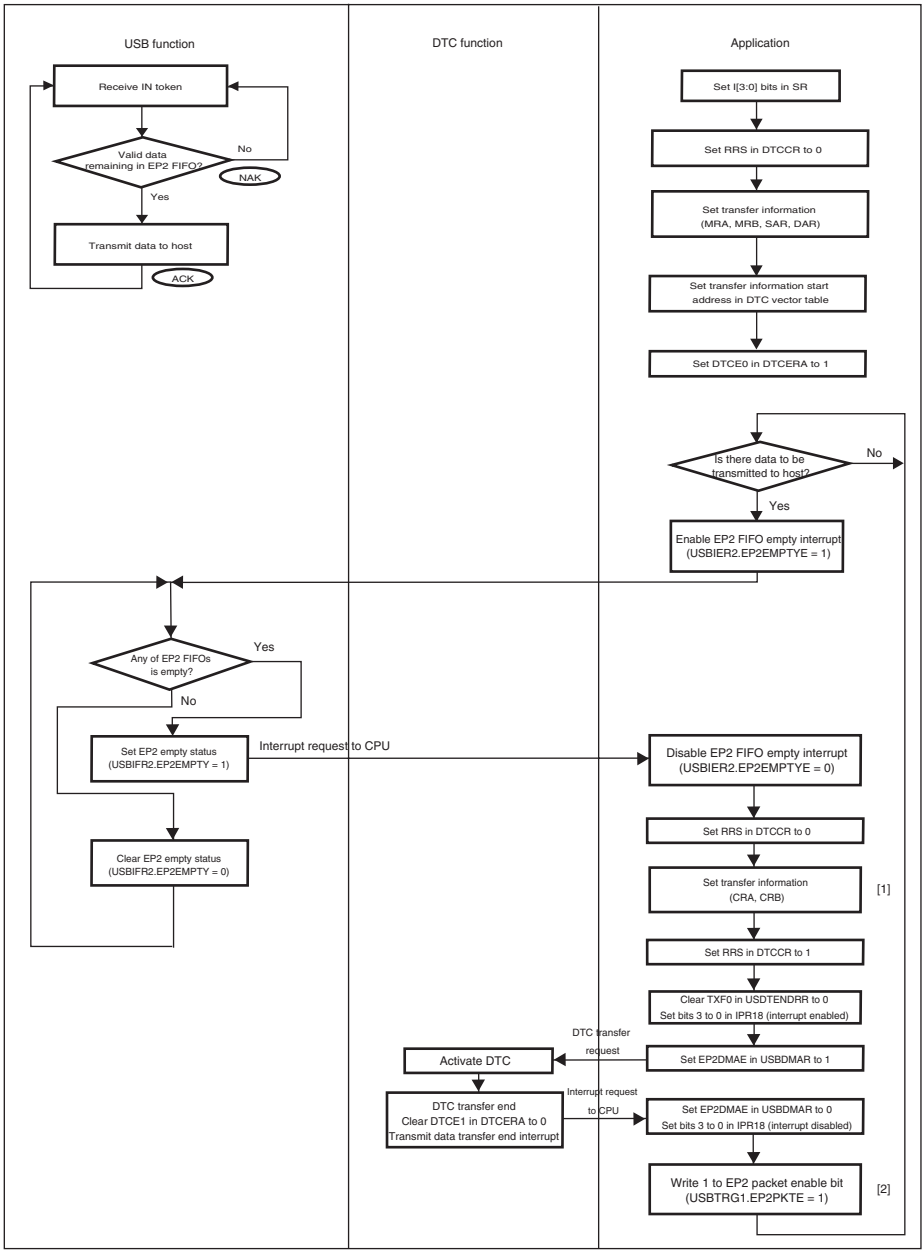
Figure 24.25 EP2/EP5 PKTE (EP2PKTE, EP5PKTE) Operation



[1] For block transfer mode, a block size of 64 bytes or less must be set in CRA.

[2] Not necessary when transmit data size is a multiple of 64 bytes.

Figure 24.26 Example of DTC Transfer for Bulk-IN Transfer (EP2) (When Transmit Data Size is Determined Before Receiving IN Token)



[1] For block transfer mode, a block size of 64 bytes or less must be set in CRA.
[2] Not necessary when transmit data size is a multiple of 64 bytes.

Figure 24.27 Example of DTC Transfer for Bulk-IN Transfer (EP2)
(When Transmit Data Size Cannot be Determined Before Receiving IN Token)

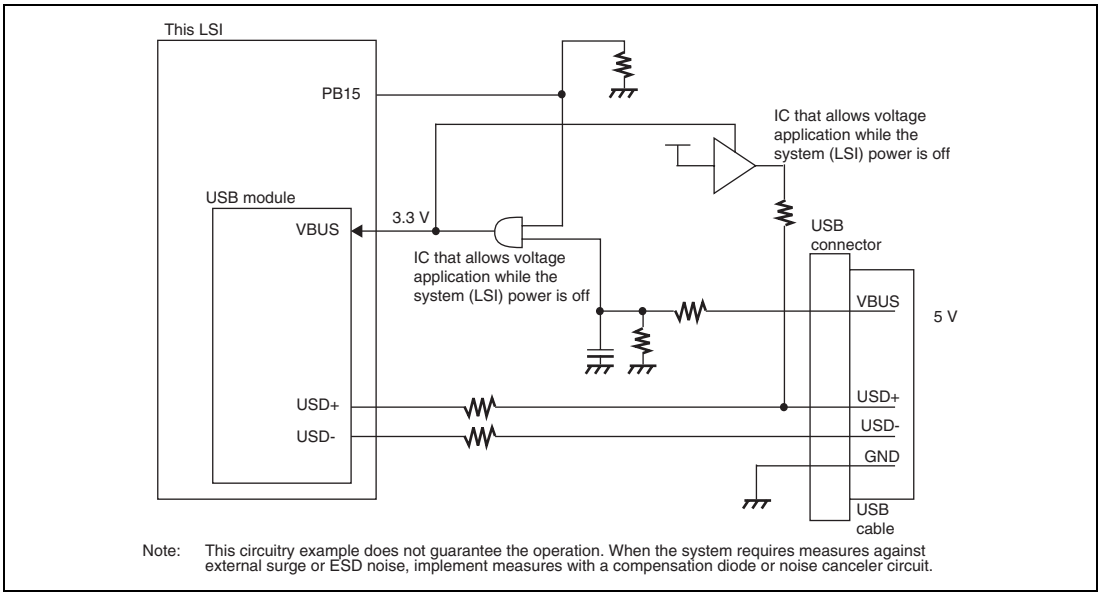
24.10 Example of USB External Circuitry

(1) USD+ Pull-Up Control

In a system that wishes to delay USB host/hub connection notification (USD+ pull-up) (during high-priority processing or initialization processing, for example), USD+ pull-up should be controlled using a general output port. When the USB cable is already connected to the host or hub and USD+ pull-up is inhibited, the USD+ and USD– signals are driven low (these signals are pulled down on the host or hub side) and the USB module incorrectly recognizes that it has received the USB bus reset signal from the host. In that case, the USD+ pull-up control signal and VBUS pin input signal should be controlled using a general output port and the USB cable VBUS (AND circuit) as shown in figure 24.28. (The UDC core of this LSI holds the powered state while the VBUS pin level is low regardless of the USD+ and USD– state.)

(2) Detection of USB Cable Connection/Disconnection

As USB states are managed by hardware in this module, a VBUS signal that recognizes USB cable connection/disconnection is necessary. The power supply signal (VBUS) in the USB cable is used for this purpose. However, if the cable is connected to the USB host or hub while the on-chip function LSI power is off, a voltage (5 V) will be applied from the USB host/hub. Therefore, an IC (HD74LV1G08A, 2G08A, etc.) that allows voltage application when the system power is off should be connected externally.



Note: This circuitry example does not guarantee the operation. When the system requires measures against external surge or ESD noise, implement measures with a compensation diode or noise canceler circuit.

Figure 24.28 Example of USB Function Module External Circuitry

24.11 Usage Notes

24.11.1 Receiving Setup Data

For USBEPDR0s that receives 8-byte setup data, note the following:

1. Since the USB always receives the setup command, writing from the USB bus has priority over reading from the CPU. When the USB starts receiving the next setup command while the CPU is reading data after data reception, the USB forcibly invalidates reading from the CPU to start writing. Therefore, the value that is read after starting reception is undefined.
2. USBEPDR0s must be read in 8-byte units. When reading is stopped in the middle, the data that is received by the next setup command cannot be read correctly.

24.11.2 Clearing FIFO

If the connected USB cable is disconnected during communication, the data being received or transmitted may remain in the FIFO. Therefore, clear the FIFO immediately after the USB cable is connected.

Do not clear the FIFO that is receiving data from or transmitting data to the host.

24.11.3 Overreading or Overwriting Data Registers

Note the following when reading or writing the data registers of this module:

(1) Receive Data Register

Do not read data that exceeds the valid receive data size from the receive data register. That is, data that exceeds the number of bytes specified in the receive data size register must not be read. For USBEPDR1 and USBEPDR4 that have two FIFOs, the maximum number of bytes that can be read at one time is 64 bytes. After reading data on the currently selected side, write 1 to the EPxRDFN bit in USBTRGx to change the current side to another side. This allows the number of bytes for the new side to be used as the receive data size, enabling the next data to be read.

(2) Transmit Data Register

Do not write data that exceeds the maximum packet size to the transmit data register. For USBEPDR2 and USBEPDR5 that have two FIFOs, the data to be written at one time must be the maximum packet size or less. After writing data, write 1 to the EPxPKTE bit in USBTRGx to change the currently selected side to another in the module to allow the next data to be written to the new side. Therefore, do not write data to one side of FIFO right after the other side.

24.11.4 Assigning Interrupt Sources for EP0

Interrupt sources (bits 0 to 3) for EP0 that are assigned to USBIFR1 of this module must be assigned to the vector number of the same interrupt request using USBISR1. There are no restrictions on other interrupt sources.

24.11.5 Clearing FIFO when Setting DMAC/DTC Transfer

When DMA/DTC transfer is enabled (USBDMAR.EP1DMAE = 1 or EP4DMAE = 1) for endpoint 1 or 4, USBEPDR1 or USBEPDR4 cannot be cleared. To clear these registers, cancel DMA/DTC transfer.

24.11.6 Manual Reset for DMAC/DTC Transfer

Do not input a manual reset during DMA/DTC transfer for endpoints 1, 2, 4, and 5. Correct operation cannot be guaranteed.

24.11.7 USB Clock

Wait for the USB clock settling time and then cancel the module stop setting for the USB function module.

24.11.8 Using TR Interrupt

Note the following when using the transfer request interrupt (TR interrupt) for interrupt-IN transfer of EP0i/EP2/EP3/EP5/EP6/EP8/EP9.

The TR interrupt flag is set when an IN token is sent from the USB host and there is no data in the FIFO of the EP. However, TR interrupts occur continuously at the timing shown in figure 24.29. Make sure that no malfunction occurs in these cases.

Note: This module checks NAK acknowledgement if there is no data in the FIFO of the EP when receiving an IN token. However, the TR interrupt flag is set after the NAK handshake is transmitted. Therefore, when writing the PKTE bit in USBTRG is later than the next IN token, the TR interrupt flag is set again.

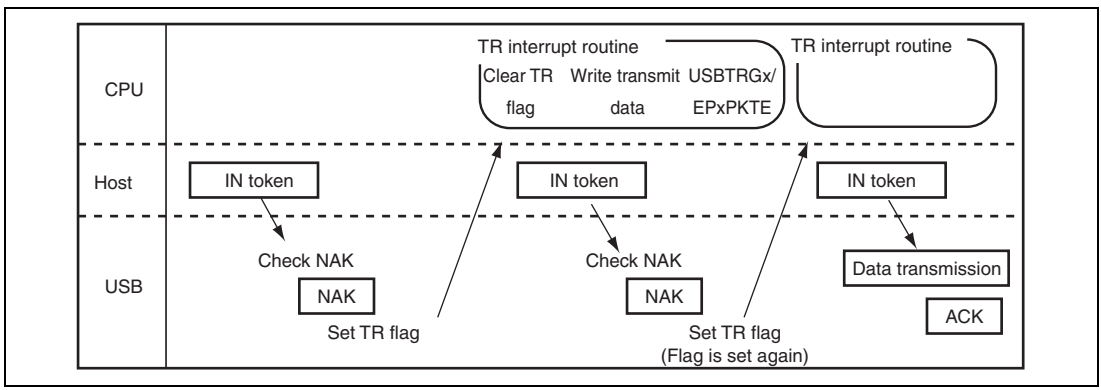


Figure 24.29 Timing for Setting the TR Interrupt Flag

24.11.9 Handling of Unused USB Pins

Handles the pins as listed below.

- DrVcc = VccQ = 3.0 to 3.6 V
- DrVss = 0 V
- USD+ = Open
- USD- = Open
- VBUS = 0 V
- USBEXTAL = 0 V
- USBXTAL = Open

Section 25 Ethernet Controller (EtherC) (SH7216A, SH7214A, SH7216G, and SH7214G only)

This LSI has an on-chip Ethernet controller (EtherC) conforming to the Ethernet or the IEEE802.3 MAC (Media Access Control) layer standard. Connecting a physical-layer LSI (PHY-LSI) conforming to this standard enables the EtherC to transmit and receive Ethernet/IEEE802.3 frames. The EtherC has one MAC layer interface port. The EtherC is connected to the Ethernet Direct Memory Access Controller (E-DMAC) for Ethernet controller inside the LSI, and carries out high-speed data transfer to and from the memory.

Figure 25.1 shows a configuration of the EtherC.

25.1 Features

- Transmission and reception of Ethernet/IEEE802.3 frames
- Supports 10/100 Mbps data transfer
- Supports full-duplex and half-duplex modes
- Conforms to IEEE802.3u standard MII (Media Independent Interface)
- Magic Packet detection and Wake-On-LAN (WOL) signal output
- Conforms to IEEE802.3x flow control

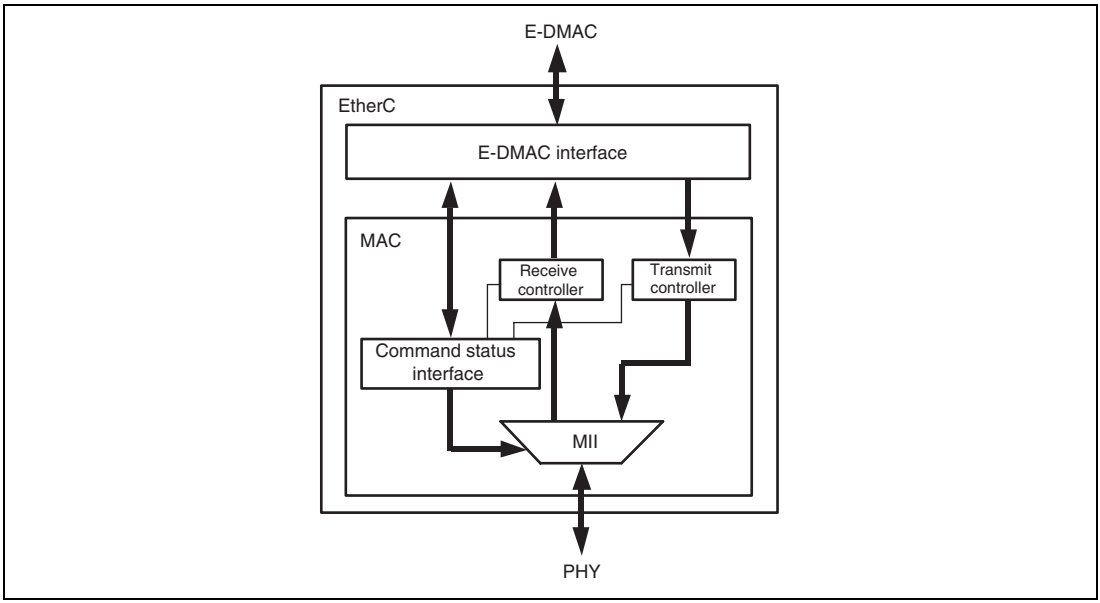


Figure 25.1 Configuration of EtherC

25.2 Input/Output Pins

Table 25.1 lists the pin configuration of the EtherC.

Table 25.1 Pin Configuration

Name	Abbreviation	I/O	Function
Transmit clock*	TX-CLK	Input	TX-EN, MII_TXD3 to MII_TXD0, TX-ER timing reference signal
Receive clock*	RX-CLK	Input	RX-DV, MII_RXD3 to MII_RXD0, RX-ER timing reference signal
Transmit enable*	TX-EN	Output	Indicates that transmit data is ready on MII_TXD3 to MII_TXD0
Transmit data*	MII_TXD3 to MII_TXD0	Output	4-bit transmit data
Transmit error*	TX-ER	Output	Notifies PHY_LSI of error during transmission
Receive data valid*	RX-DV	Input	Indicates that valid receive data is present on MII_RXD3 to MII_RXD0
Receive data*	MII_RXD3 to MII_RXD0	Input	4-bit receive data
Receive error*	RX-ER	Input	Identifies error state occurred during data reception
Carrier detection*	CRS	Input	Carrier detection signal
Collision detection*	COL	Input	Collision detection signal
Management data clock*	MDC	Output	Reference clock signal for information transfer via MDIO
Management data I/O*	MDIO	I/O	Bidirectional signal to exchange management information between STA and PHY
Link status	LNKSTA	Input	Inputs link status from PHY
General-purpose external output	EXOUT	Output	External output pin
Wake-On-LAN	WOL	Output	Indicates reception of Magic Packet

Note: * MII signal conforming to IEEE802.3u

25.3 Register Descriptions

Table 25.2 shows the configuration of registers of EtherC.

Table 25.2 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
EtherC mode register	ECMR	R/W	H'FFFC 3100	32
EtherC status register	ECSR	R/W	H'FFFC 3110	32
EtherC interrupt enable register	ECSIPR	R/W	H'FFFC 3118	32
Receive frame length register	RFLR	R/W	H'FFFC 3108	32
PHY interface register	PIR	R/W	H'FFFC 3120	32
MAC address high register	MAHR	R/W	H'FFFC 31C0	32
MAC address low register	MALR	R/W	H'FFFC 31C8	32
PHY status register	PSR	R	H'FFFC 3128	32
Transmit retry over counter register	TROCR	R/W	H'FFFC 31D0	32
Delayed collision detect counter register	CDCR	R/W	H'FFFC 31D4	32
Lost carrier counter register	LCCR	R/W	H'FFFC 31D8	32
Carrier not detect counter register	CNDCR	R/W	H'FFFC 31DC	32
CRC error frame receive counter register	CEFCR	R/W	H'FFFC 31E4	32
Frame receive error counter register	FRECR	R/W	H'FFFC 31E8	32
Too-short frame receive counter register	TSFRCR	R/W	H'FFFC 31EC	32
Too-long frame receive counter register	TLFRCR	R/W	H'FFFC 31F0	32
Residual-bit frame receive counter register	RFCR	R/W	H'FFFC 31F4	32
Multicast address frame receive counter register	MAFCR	R/W	H'FFFC 31F8	32
IPG register	IPGR	R/W	H'FFFC 3150	32
Automatic PAUSE frame register	APR	R/W	H'FFFC 3154	32
Manual PAUSE frame register	MPR	R/W	H'FFFC 3158	32
Automatic PAUSE frame retransmit count register	TPAUSER	R/W	H'FFFC 3164	32
Random number generation counter upper limit register	RDMLR	R/W	H'FFFC 3140	32

Name	Abbreviation	R/W	Address	Access Size
PAUSE frame receive counter register	RFCF	R/W	H'FFFC 3160	32
PAUSE frame retransmit counter register	TPAUSECR	R	H'FFFC 3168	32
Broadcast frame receive count register	BCFRR	R	H'FFFC 316C	32

Table 25.3 shows the EtherC register status in each operating mode.

Table 25.3 Register States in Each Operating Mode

Name	Abbreviation	Software Reset
EtherC mode register	ECMR	Initialized
EtherC status register	ECSR	Initialized
EtherC interrupt enable register	ECSIPR	Initialized
Receive frame length register	RFLR	Initialized
PHY interface register	PIR	Initialized
MAC address high register	MAHR	Initialized
MAC address low register	MALR	Initialized
PHY status register	PSR	Initialized
Transmit retry over counter register	TROCR	Initialized
Delayed collision detect counter register	CDCR	Initialized
Lost carrier counter register	LCCR	Initialized
Carrier not detect counter register	CNDCR	Initialized
CRC error frame receive counter register	CEFCR	Initialized
Frame receive error counter register	FRECR	Initialized
Too-short frame receive counter register	TSFRCR	Initialized
Too-long frame receive counter register	TLFRCR	Initialized
Residual-bit frame receive counter register	RFCR	Initialized
Multicast address frame receive counter register	MAFCR	Initialized
IPG register	IPGR	Initialized
Automatic PAUSE frame register	APR	Initialized
Manual PAUSE frame register	MPR	Initialized
Automatic PAUSE frame retransmit count register	TPAUSER	Initialized
Random number generation counter upper limit register	RDMLR	Initialized
PAUSE frame receive counter register	RFCF	Initialized
PAUSE frame retransmit counter register	TPAUSECR	Initialized
Broadcast frame receive count register	BCFRR	Initialized

25.3.1 EtherC Mode Register (ECMR)

ECMR is a 32-bit readable/writable register that specifies the operating mode of the EtherC. The settings of this register are normally made in the initialization process after a reset.

The operating mode setting must not be changed while the transmitting and receiving functions are enabled. To change the operating mode, return the EtherC and E-DMAC to their initial states with the SWR bit in EDMR of the E-DMAC before making settings again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	TPC	ZPF	PFR	RXF	TXF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	PRCEF	–	–	MPDE	–	–	RE	TE	–	ILB	–	DM	PRM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R	R	R/W	R/W	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	TPC	0	R/W	PAUSE Frame Transmission 0: PAUSE frame is not transmitted in a PAUSE period 1: PAUSE frame is transmitted even in a PAUSE period
19	ZPF	0	R/W	PAUSE Frame Usage with TIME = 0 Enable 0: Control of a PAUSE frame whose TIME parameter value is 0 is disabled. The next frame is not transmitted until the time specified by the Timer value has elapsed. If a PAUSE frame whose time specified by the Timer value is 0 is received, the PAUSE frame is discarded. 1: Control of a PAUSE frame whose TIME parameter value is 0 is enabled. When the data size in the receive FIFO becomes smaller than the FCFTR setting 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.

Bit	Bit Name	Initial Value	R/W	Description
18	PFR	0	R/W	PAUSE Frame Receive Mode 0: PAUSE frame is not transferred to the E-DMAC 1: PAUSE frame is transferred to the E-DMAC
17	RXF	0	R/W	Operating Mode for Receiving Port Flow Control 0: PAUSE frame detection is disabled 1: The receiving port flow control is enabled
16	TXF	0	R/W	Operating Mode for Transmitting Port Flow Control 0: PAUSE frame detection is disabled (Automatic PAUSE frame is not transmitted) 1: The transmitting port flow control is enabled (Automatic PAUSE frame is transmitted as required)
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	PRCEF	0	R/W	CRC Error Frame Reception Enable 0: A receive frame with a CRC error is treated as an error frame 1: A receive frame with a CRC error is not treated as an error frame
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	MPDE	0	R/W	Magic Packet Detection Enable Enables or disables Magic Packet detection by hardware to allow activation from the Ethernet. 0: Magic Packet detection is disabled 1: Magic Packet detection is enabled
8, 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	RE	0	R/W	Reception Enable When this bit is changed from RE = 1 (receiving function enabled) to RE = 0 (disabled) while a frame is being received, the receiving function will be enabled until the frame reception is completed. 0: Receiving function is disabled 1: Receiving function is enabled
5	TE	0	R/W	Transmission Enable When this bit is changed from TE = 1 (transmitting function enabled) to TE = 0 (disabled) while a frame is being transmitted, the transmitting function will be enabled until the frame transmission is completed. 0: Transmitting function is disabled 1: Transmitting function is enabled
4	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
3	ILB	0	R/W	Internal Loopback Mode Specifies loopback mode in the EtherC. 0: Normal data transmission/reception is performed 1: Data is looped back inside the MAC in the EtherC when DM = 1
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	DM	0	R/W	Duplex Mode Specifies the EtherC transfer method. 0: Half-duplex transfer is specified 1: Full-duplex transfer is specified

Bit	Bit Name	Initial Value	R/W	Description
0	PRM	0	R/W	Promiscuous Mode Setting this bit to 1 enables all Ethernet frames to be received, that is, all receivable frames regardless of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.). 0: The EtherC performs normal operation 1: The EtherC performs promiscuous mode operation

25.3.2 EtherC Status Register (ECSR)

ECSR is a 32-bit readable/writable register that indicates the status in the EtherC. This status 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 to 0. Writing 0 does not affect any flags. For bits that generate interrupts, the interrupt can be enabled or disabled by the corresponding bit in ECSIPR.

The interrupts generated due to this status register are reflected in the ECI bit in EESR of the E-DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	BFR	PSRTO	–	LCHNG	MPD	ICD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	BFR	0	R/W	Continuous Broadcast Frame Reception Interrupt (Interrupt Source) Indicates that broadcast frames have been continuously received.
4	PSRTO	0	R/W	PAUSE Frame Retransmit Retry Over Indicates whether the retransmit count for retransmitting a PAUSE frame when flow control is enabled has exceeded the retransmit upper-limit value set in the automatic PAUSE frame retransmit count register (TPAUSER). 0: PAUSE frame retransmit count has not exceeded the upper limit 1: PAUSE frame retransmit count has exceeded the upper limit

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	LCHNG	0	R/W	Link Signal Change Indicates that the LNKSTA 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 LMON bit in the PHY status register (PSR). 0: A change in the LNKSTA signal has not been detected 1: A change in the LNKSTA signal has been detected (high to low or low to high)
1	MPD	0	R/W	Magic Packet Detection Indicates that a Magic Packet has been detected on the line. 0: No Magic Packet has been detected 1: A Magic Packet has been detected
0	ICD	0	R/W	Illegal Carrier Detection Indicates that the PHY-LSI has detected an illegal carrier on the line. More specifically, this bit is set to 1 when the signals transmitted from the PHY-LSI to this LSI become RX-DV = 0, RX-ER = 1, and MII-RXD3 to MII-RXD0 = 1110 (see figure 25.4 (6)). If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, correct information may not be obtained. Refer to the timing specification for the PHY-LSI used. 0: The PHY-LSI has not detected an illegal carrier on the line 1: The PHY-LSI has detected an illegal carrier on the line

25.3.3 EtherC Interrupt Enable Register (ECSIPR)

ECSIPR is a 32-bit readable/writable register that enables or disables the interrupt sources indicated in ECSR. Each bit in ECSIPR can enable or disable interrupts corresponding to the bits in ECSR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	BFSIPR	PSRTOIP	–	LCHNGIP	MPDIP	ICDIP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	BFSIPR	0	R/W	Continuous Broadcast Frame Reception Interrupt Enable 0: Enables an interrupt requested by the BFR bit in ECSR 1: Disables an interrupt requested by the BFR bit in ECSR
4	PSRTOIP	0	R/W	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
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2	LCHNGIP	0	R/W	LINK Signal Change Interrupt Enable 0: Interrupt notification by the LCHNG bit is disabled 1: Interrupt notification by the LCHNG bit is enabled
1	MPDIP	0	R/W	Magic Packet Detect Interrupt Enable 0: Interrupt notification by the MPD bit is disabled 1: Interrupt notification by the MPD bit is enabled
0	ICDIP	0	R/W	Illegal Carrier Detect Interrupt Enable 0: Interrupt notification by the ICD bit is disabled 1: Interrupt notification by the ICD bit is enabled

25.3.4 PHY Interface Register (PIR)

PIR is a 32-bit readable/writable register that provides a means of accessing the PHY-LSI internal registers through the MII.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	MDI	MDO	MMD	MDC
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	Undefined	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MDI	Undefined	R	MII Management Data-In Indicates the MDIO pin level.
2	MDO	0	R/W	MII Management Data-Out Outputs the value of this bit from the MDIO pin when the MMD bit is 1.
1	MMD	0	R/W	MII Management Mode Specifies the direction of data read from/data write to the MII. 0: Read direction is specified 1: Write direction is specified
0	MDC	0	R/W	MII Management Data Clock Outputs the value of this bit from the MDC pin to supply the MII with the management data clock. For how to access the MII registers, see section 25.4.4, Accessing MII Registers.

25.3.5 MAC Address High Register (MAHR)

MAHR is a 32-bit readable/writable register that specifies the upper 32 bits of 48-bit MAC address. This register is normally set in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Reset the EtherC and E-DMAC with the SWR bit in EDMR of the E-DMAC, and then set the MAC address again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MA[47:32]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MA[47:16]	All 0	R/W	MAC Address Bits 47 to 16 These bits are used to set the upper 32 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'01234567 in this register.

25.3.6 MAC Address Low Register (MALR)

MALR is a 32-bit readable/writable register that specifies the lower 16 bits of 48-bit MAC address. This register is normally set in the initialization process after a reset. The MAC address setting must not be changed while the transmitting and receiving functions are enabled. Reset the EtherC and E-DMAC with the SWR bit in EDMR of the E-DMAC, and then set the MAC address again.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MA[15:0]	All 0	R/W	MAC Address Bits 15 to 0 These bits are used to set the lower 16 bits of the MAC address. If the MAC address is 01-23-45-67-89-AB (hexadecimal), set H'89AB in this register.

25.3.7 Receive Frame Length Register (RFLR)

RFLR is a 32-bit readable/writable register that specifies the maximum frame length (in bytes) that can be received by this LSI. This register must not be modified while the receiving function is enabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	RFL[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	RFL[11:0]	All 0	R/W	Receive Frame Data Length The frame data described here refers to all fields from the destination address up to the CRC data. Frame content from the destination address up to the data (excluding CRC data) is actually transferred to the memory. When data that exceeds the value of these bits is received, the excess part of the data is discarded. H'000 to H'5EE: 1,518 bytes H'5EF: 1,519 bytes H'5F0: 1,520 bytes : : H'7FF: 2,047 bytes H'800 to H'FFF: 2048 bytes

25.3.8 PHY Status Register (PSR)

PSR is a read-only register that can read the interface signal from the PHY-LSI.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	LMON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Undefined
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	LMON	Undefined	R	LNKSTA Pin Status The Link status can be read by connecting the Link signal output from the PHY-LSI to the LNKSTA pin. For the signal polarity, refer to the specifications of the PHY-LSI to be connected.

25.3.9 Transmit Retry Over Counter Register (TROCR)

TROCR is a 32-bit counter that indicates the number of frames that were not transmitted in 16 transmission attempts including retransmission. When transmission fails 16 times, this register value is incremented by 1. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TROC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TROC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TROC[31:0]	All 0	R/W	Transmit Retry Over Count These bits indicate the number of frames that were not transmitted in 16 transmission attempts including retransfer.

25.3.10 Delayed Collision Detect Counter Register (CDCR)

CDCR is a 32-bit counter that indicates the number of all delayed collisions that occurred on the line from the beginning of data transmission. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COSDC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COSDC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	COSDC[31:0]	All 0	R/W	Delayed Collision Detect Count These bits indicate the number of all delayed collisions occurred from the beginning of data transmission.

25.3.11 Lost Carrier Counter Register (LCCR)

LCCR is a 32-bit counter that indicates the number of times the carrier was lost during data transmission. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LCC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LCC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	LCC[31:0]	All 0	R/W	Lost Carrier Count These bits indicate the number of times the carrier was lost during data transmission.

25.3.12 Carrier Not Detect Counter Register (CNDCR)

CNDCR is a 32-bit counter that indicates the number of times the carrier was not detected during transmission of the preamble. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNDC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNDC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CNDC[31:0]	All 0	R/W	Carrier Not Detect Count These bits indicate the number of times the carrier was not detected.

25.3.13 CRC Error Frame Receive Counter Register (CEFCR)

CEFCR is a 32-bit counter that indicates the number of times a frame with a CRC error was received. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CEFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CEFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	CEFC[31:0]	All 0	R/W	CRC Error Frame Count These bits indicate the number of CRC error frames received.

25.3.14 Frame Receive Error Counter Register (FRECR)

FRECR is a 32-bit counter that indicates the number of frames in which a receive error was generated by the RX-ER signal input from the PHY-LSI. FRECR is incremented each time the RX-ER pin becomes active. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FREC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FREC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FREC[31:0]	All 0	R/W	Frame Receive Error Count These bits indicate the number of errors occurred during frame reception.

25.3.15 Too-Short Frame Receive Counter Register (TSFRCR)

TSFRCR is a 32-bit counter that indicates the number of frames received with a length of less than 64 bytes. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TSFC[31:0]	All 0	R/W	Too-Short Frame Receive Count These bits indicate the number of too-short (less than 64 bytes) frames received.

25.3.16 Too-Long Frame Receive Counter Register (TLFRCR)

TLFRCR is a 32-bit counter that indicates the number of frames received with a length exceeding the value specified by the receive frame length register (RFLR). When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. This register is not incremented when a frame containing residual bits is received. In this case, the reception of the frame is reflected in the residual-bit frame receive counter register (RFCR). The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TLFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TLFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TLFC[31:0]	All 0	R/W	Too-Long Frame Receive Count These bits indicate the number of too-long (exceeding the RFLR value) frames received.

25.3.17 Residual-Bit Frame Receive Counter Register (RFCR)

RFCR is a 32-bit counter that indicates the number of frames received containing residual bits (less than an 8-bit unit). When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RFC[31:0]	All 0	R/W	Residual-Bit Frame Receive Count These bits indicate the number of frames received containing residual bits.

25.3.18 Multicast Address Frame Receive Counter Register (MAFCR)

MAFCR is a 32-bit counter that indicates the number of received frames that specify a multicast address. When the value of this register reaches H'FFFFFFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MAFC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MAFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MAFC[31:0]	All 0	R/W	Multicast Address Frame Count These bits indicate the number of multicast address frames received.

25.3.19 IPG Register (IPGR)

IPGR is used to set an IPG (Inter Packet Gap) value. This register must not be modified while the transmitting and receiving functions of the EtherC mode register (ECMR) are enabled. (For details, see section 25.4.6, Operation by IPG Setting.)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	IPG[4:0]				
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	IPG[4:0]	H'14	R/W	Inter Packet Gap An IPG value is set in units of 4-bit time. H'00: 16-bit time H'01: 20-bit time : : H'14: 96-bit time (default) : : H'1F: 140-bit time

25.3.20 Automatic PAUSE Frame Register (APR)

APR is used to set the TIME parameter value of an automatic PAUSE frame. When an automatic PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AP[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	AP[15:0]	All 0	R/W	Automatic PAUSE These bits set the TIME parameter value of an automatic PAUSE frame. One bit is equivalent to 512-bit time.

25.3.21 Manual PAUSE Frame Register (MPR)

MPR is used to set the TIME parameter value of a manual PAUSE frame. When a manual PAUSE frame is transmitted, the value set in this register is used as the TIME parameter of the PAUSE frame.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MP[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MP[15:0]	All 0	R/W	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame. One bit is equivalent to 512-bit time. Read value is undefined.

25.3.22 Automatic PAUSE Frame Retransmit Count Register (TPAUSER)

TPAUSER is used to set the upper limit for the number of times to retransmit an automatic PAUSE frame. This register must not be modified while the transmitting function is enabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TPAUSE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	TPAUSE[15:0]	All 0	R/W	Upper Limit for Automatic PAUSE Frame Retransmission Count H'0000: Retransmit count is unlimited H'0001: Retransmit count is 1 : : H'FFFF: Retransmit count is 65,535

25.3.23 Random Number Generation Counter Upper Limit Register (RDMLR)

RDMLR is used to set the upper limit for the counter used in the random number generation block.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	RMD[19:16]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMD[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 0	RMD[19:0]	All 0	R/W	Upper Limit for Counter Used in Random Number Generation Block H'00000: Used in normal operation H'00001to H'FFFFE: Upper limit for the counter

Note: The setting of this register affects the operation of the random number generation block in the feLic. Pay attention when setting a value other than H'00000.

25.3.24 PAUSE Frame Receive Counter Register (RFCF)

RFCF is a counter that indicates the number of times a PAUSE frame is received.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	RPAUSE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	RPAUSE[7:0]	All 0	R	PAUSE Frame Receive Count

25.3.25 PAUSE Frame Retransmit Counter Register (TPAUSECR)

TPAUSECR is a counter that indicates the number of times a PAUSE frame is retransmitted.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	TXP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	TXP[7:0]	All 0	R	PAUSE Frame Retransmit Count

25.3.26 Broadcast Frame Receive Count Register (BCFRR)

BCFRR is used to set the number of broadcast frames that can be received continuously.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	BCF[15:0]	All 0	R/W	Receive Count for Continuous Broadcast Frames The DA can receive a broadcast address frame up to the number of times set in these bits. If broadcast address frames are received more often than the set value, the excess frames are discarded. H'0000: Receive count is unlimited H'0001: 1 frame can be received : : H'FFFF: 65,535 continuous frames can be received

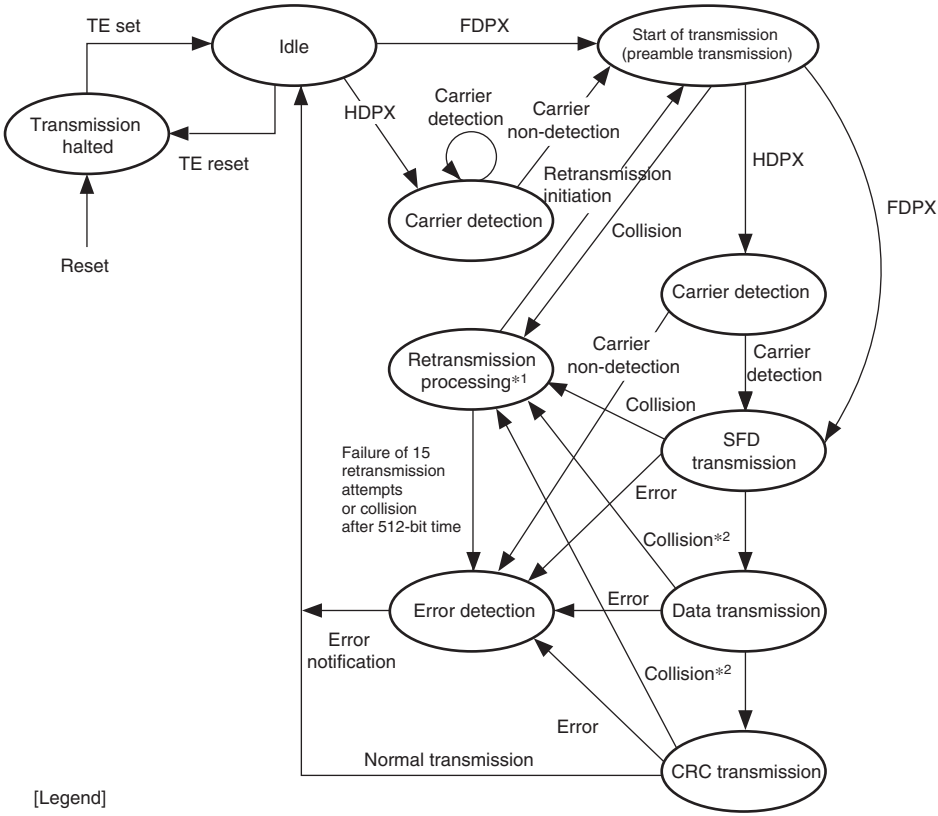
25.4 Operation

The following outlines the operations of the Ethernet controller (EtherC).

The EtherC supports control functions conforming to IEEE802.3x, allowing transmission/reception of PAUSE frames used for the control.

25.4.1 Transmission

The EtherC transmitter assembles transmit data into a frame and outputs it to the MII when a transmit request is made from the E-DMAC. The data transferred through the MII is output to the line by the PHY-LSI. Figure 25.2 illustrates state transitions of the EtherC transmitter.



[Legend]
FDPX: Full Duplex
HDPX: Half Duplex
SFD: Start Frame Delimiter

Notes:1. Retransmission processing includes both jam transmission resultant from collision detection and the adjustment of transmission intervals by the back-off algorithm.
2. Retransmission is performed only during transmission of 512-bit (or less) data including the preamble and SFD. If a collision is detected while data exceeding 512 bits is being transmitted, only jam is transmitted and the retransmission processing by the back-off algorithm is not performed.

Figure 25.2 EtherC Transmitter State Transitions

1. When the transmit enable (TE) bit is set to 1, the transmitter enters the idle state.
2. When a transmit request is issued by the transmit E-DMAC, the EtherC detects carrier and sends the preamble after 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 E-DMAC.
3. The transmitter sends the SFD, data, and CRC sequentially. At the end of transmission, the transmit E-DMAC generates a transmission complete interrupt (TC). If a collision occurs or the carrier cannot be detected during data transmission, these events are reported as interrupt sources.
4. After the frame interval time has passed, the transmitter enters the idle state and continues to transmit data if there is more transmit data.

25.4.2 Reception

The EtherC receiver disassembles a frame sent from the MII into preamble, SFD, data, and CRC, and then transfers the fields from DA (destination address) to the CRC data to the receive E-DMAC. Figure 25.3 illustrates the state transitions of the EtherC receiver.

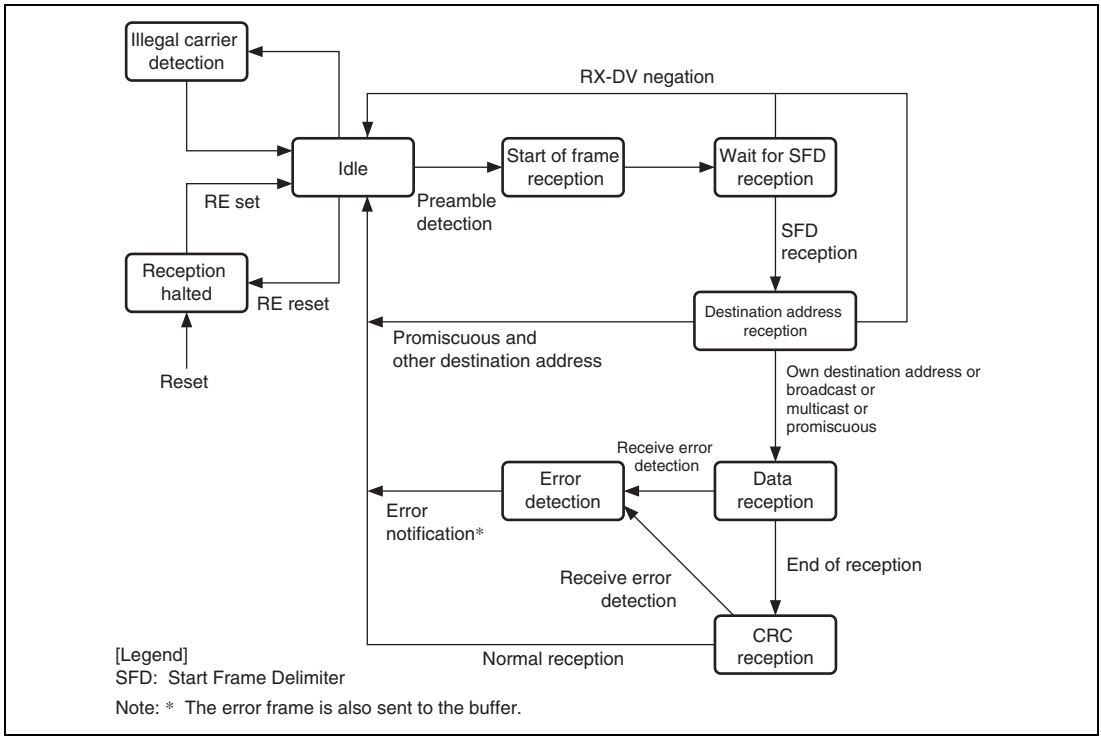


Figure 25.3 EtherC Receiver State Transitions

1. When the receive enable (RE) bit is set to 1, the receiver enters the idle state.
2. When the receiver detects an SFD (start frame delimiter) following the preamble in a receive packet, it starts receive processing. The receiver discards a frame with an invalid pattern.
3. In normal mode, if the destination address in a frame matches the address of this LSI, or if the frame is a broadcast frame or multicast frame, the receiver starts data reception. In promiscuous mode, the receiver starts data reception regardless of the frame type.
4. After receiving data from the MII, the receiver performs a CRC check. The check result is indicated as a status flag in the descriptor after the frame data has been written to the memory. The receiver reports an error status in the case of a CRC error.
5. After one frame has been received, if the receive enable bit is set (RE = 1) in the EtherC mode register, the receiver prepares to receive the next frame.

25.4.3 MII Frame Timing

Each MII frame timing is shown in figure 25.4.

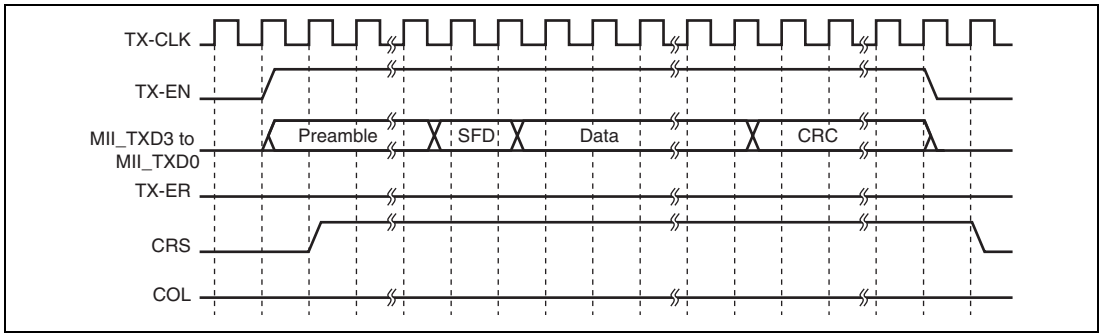


Figure 25.4 (1) MII Frame Transmit Timing (Normal Transmission)

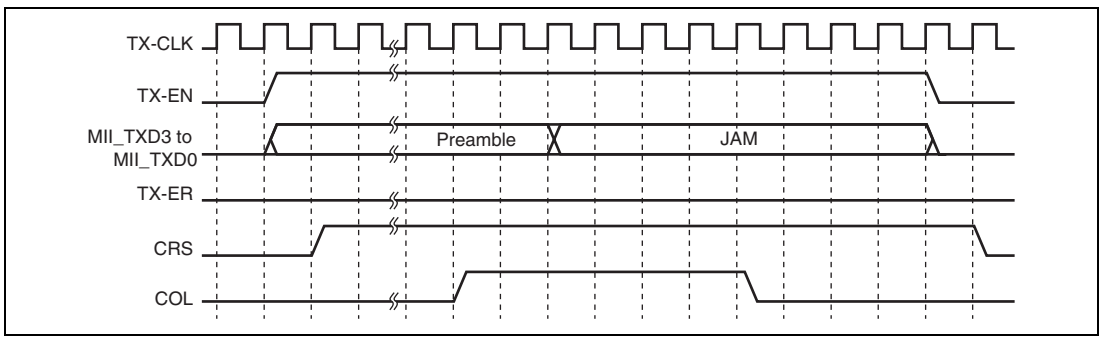


Figure 25.4 (2) MII Frame Transmit Timing (Collision)

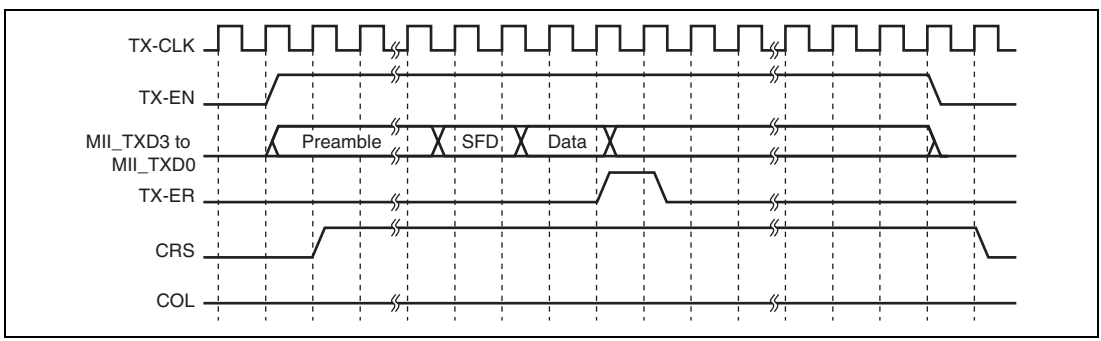


Figure 25.4 (3) MII Frame Transmit Timing (Transmit Error)

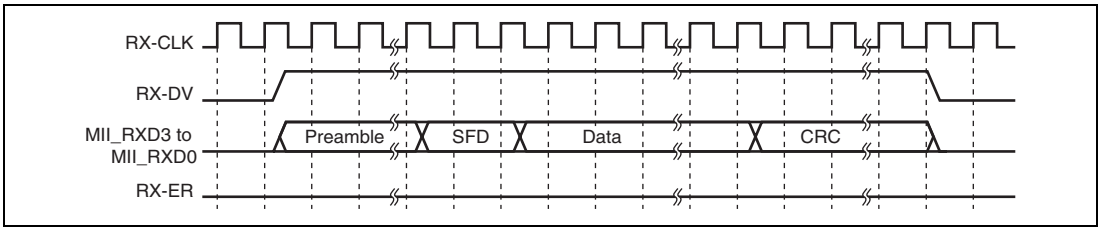


Figure 25.4 (4) MII Frame Receive Timing (Normal Reception)

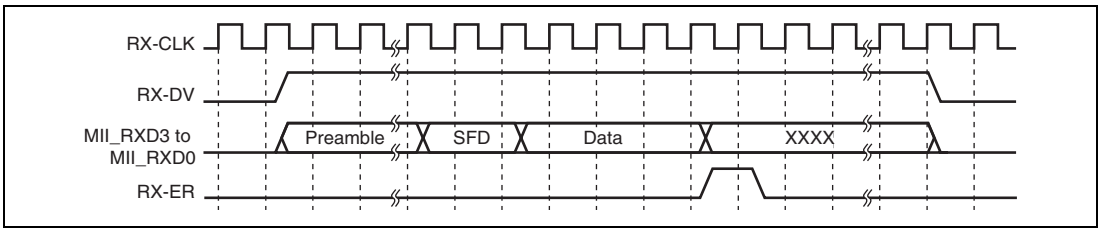


Figure 25.4 (5) MII Frame Receive Timing (Receive Error (1): Receive Error Notification)

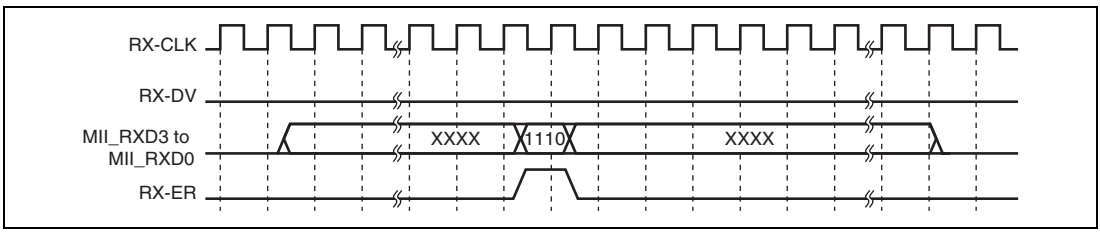


Figure 25.4 (6) MII Fame Receive Timing (Receive Error (2): Carrier Error Notification)

25.4.4 Accessing MII Registers

MII registers in the PHY-LSI are accessed through the PHY interface register (PIR) in this LSI. Connection is made as a serial interface in accordance with the MII frame format specified in IEEE802.3u.

(1) MII Management Frame Format

Figure 25.5 shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in (2) MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
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 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 when the PHY-LSI address is 1 (sequential write starting with the MSB)
The PHYAD bits vary with the PHY-LSI address.
REGAD: Write of 0001 when the register address is 1 (sequential write starting with the MSB)
The REGAD bits vary with the PHY-LSI register address.
TA: Time for switching data transmission source on the MII interface
(a) Write: 10 written
(b) Read: Bus release [notation: Z0] performed
DATA: 16-bit data. Sequential write or read starting with the MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release [notation: X] performed
(b) Read: Bus already released at TA (control unnecessary)

Figure 25.5 MII Management Frame Format

(2) MII Register Access Procedure

The program accesses MII registers through the PHY interface register (PIR). An access is made by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. Figure 25.6 shows the MII register access timing. The access timing differs depending on the PHY-LSI type.

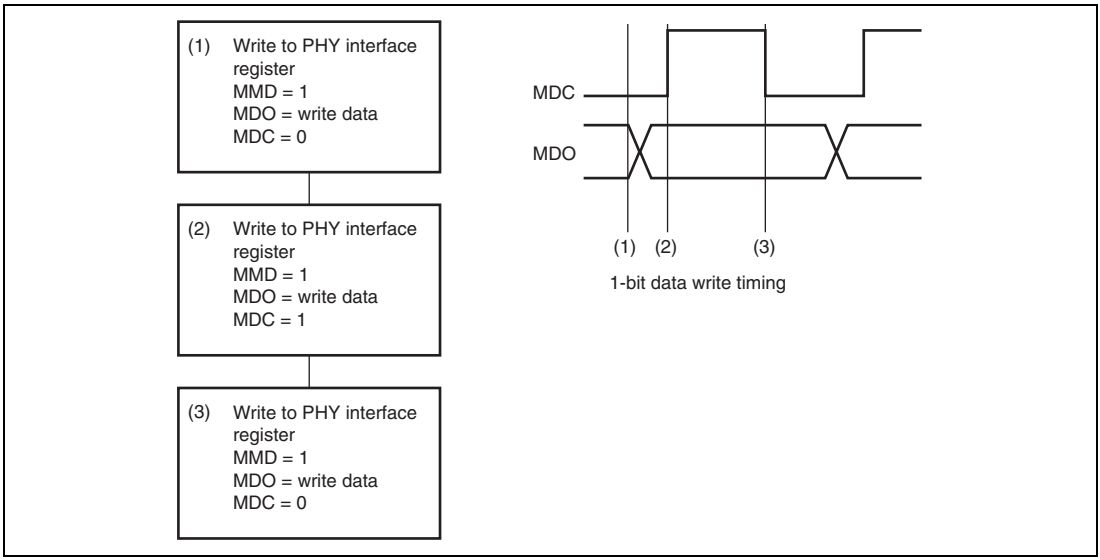


Figure 25.6 (1) 1-Bit Data Write Flow

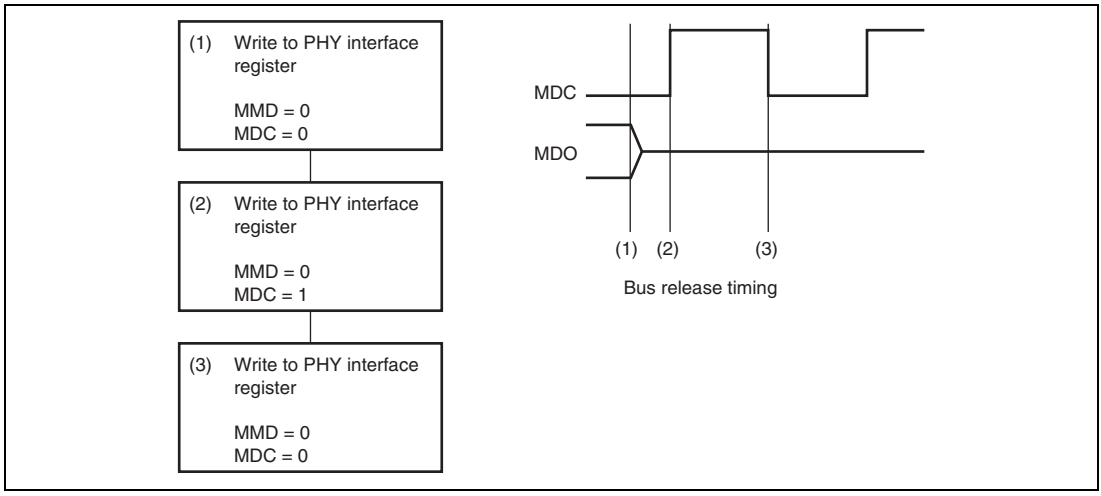


Figure 25.6 (2) Bus Release Flow (TA in Read in Figure 25.5)

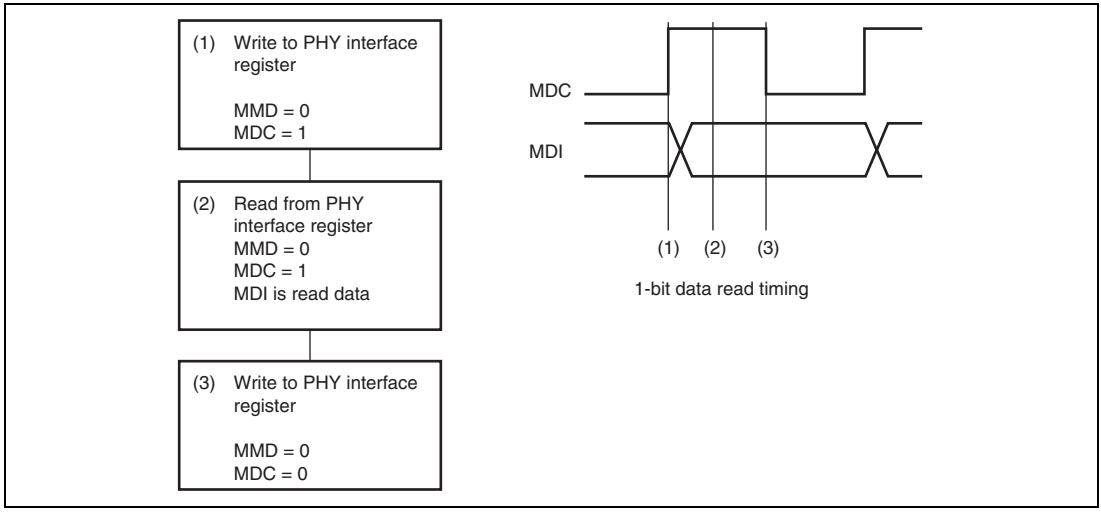


Figure 25.6 (3) 1-Bit Data Read Flow

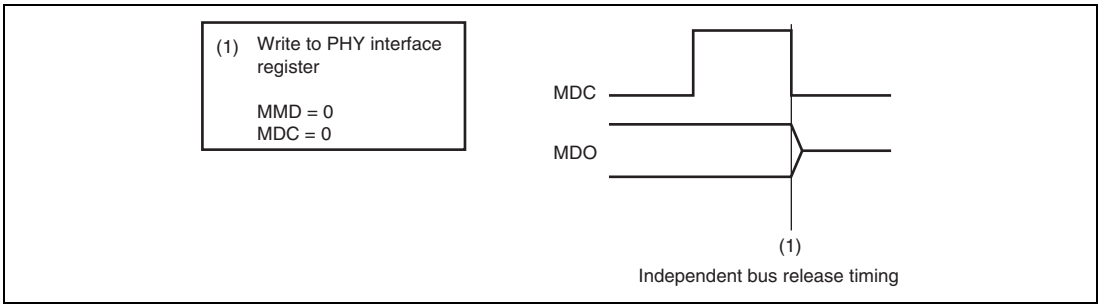


Figure 25.6 (4) Independent Bus Release Flow (IDLE in Write in Figure 25.5)

25.4.5 Magic Packet Detection

The EtherC has a Magic Packet detection function. This function provides a Wake-On-LAN (WOL) feature 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 another source, and activates itself. When the Magic Packet is detected, data (such as the broadcast packets received previously) is stored in the receive FIFO and the EtherC is notified of the receiving status. To return to normal operation from the interrupt processing, initialize the EtherC and E-DMAC with the SWR bit in the E-DMAC mode register (EDMR).

Magic Packets are received regardless of the destination address. As a result, this function and the WOL pin are enabled only when the destination address matches the address specified by the format in the Magic Packet. Further information on Magic Packets is available in the technical documentation published by AMD Corporation.

The following setting procedure is necessary to use the WOL feature with this LSI.

1. Disable interrupt source output by means of the various interrupt enable/mask registers.
2. Set the Magic Packet detection enable (MPDE) bit in the EtherC mode register (ECMR).
3. Set the Magic Packet detection interrupt enable (MPDIP) bit in the EtherC interrupt enable register (ECSIPR) to 1.
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 and the WOL pin notifies peripheral LSIs that the Magic Packet has been detected.

25.4.6 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 set value of the IPG register (IPGR), the transmission efficiency can be raised and lowered from the standard value. IPG settings are prescribed in the IEEE802.3 standard. When changing IPG settings, adequately check that the respective devices can operate smoothly on the same network.

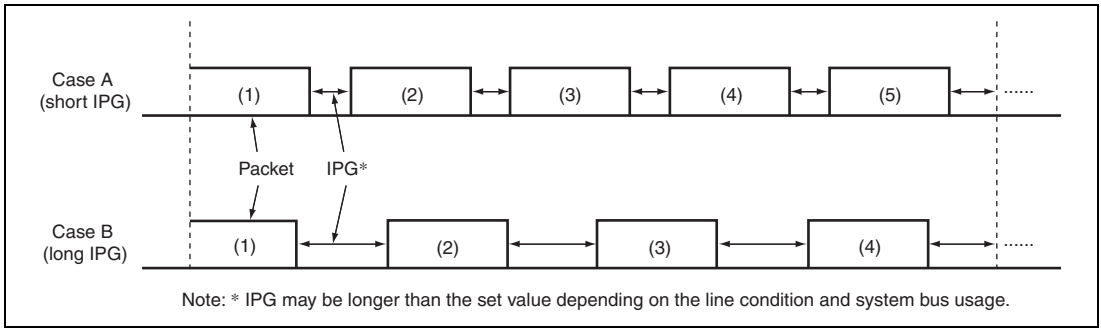


Figure 25.7 Changing IPG and Transmission Efficiency

25.4.7 Flow Control

The EtherC supports flow control functions conforming to IEEE802.3x for full-duplex operation. The flow control is available for both receive and transmit operations. When transmitting PAUSE frames, flow control can be performed in the following two procedures:

(1) Transmitting Automatic PAUSE Frames

For receive frames, PAUSE frames are automatically transmitted when the volume of data written to the receive FIFO (in the E-DMAC) reaches the value set in the flow control start FIFO threshold setting register (FCFTR) in the E-DMAC. The TIME parameter contained in the PAUSE frame is set by the automatic PAUSE frame register (APR). The automatic PAUSE frame transmission is repeated until the volume of data in the receive FIFO becomes less than the FCFTR value as the receive data is read from the FIFO. The upper limit of PAUSE frame retransmission counts can also be set in the automatic PAUSE frame retransmit count register (TPAUSER). In this case, PAUSE frame transmission is repeated until the volume of receive FIFO data becomes less than the FCFTR value, or the transmit count reaches the TPAUSER value. Transmission of automatic PAUSE frames is enabled when the TXF bit in the EtherC mode register (ECMR) is 1.

(2) Transmitting Manual PAUSE Frames

PAUSE frames are transmitted by software instructions. When a 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.

(3) Receiving PAUSE Frames

After a PAUSE frame is received, the next frame is not transmitted until the time indicated by the Timer value elapses. However, the ongoing transmission of a frame is continued. Reception of PAUSE frames is enabled when the RXF bit in ECMR is set to 1.

25.5 Connection to the PHY-LSI

Figure 25.8 shows an example of connection to the RTL8201CP (Realtek Semiconductor Corp.).

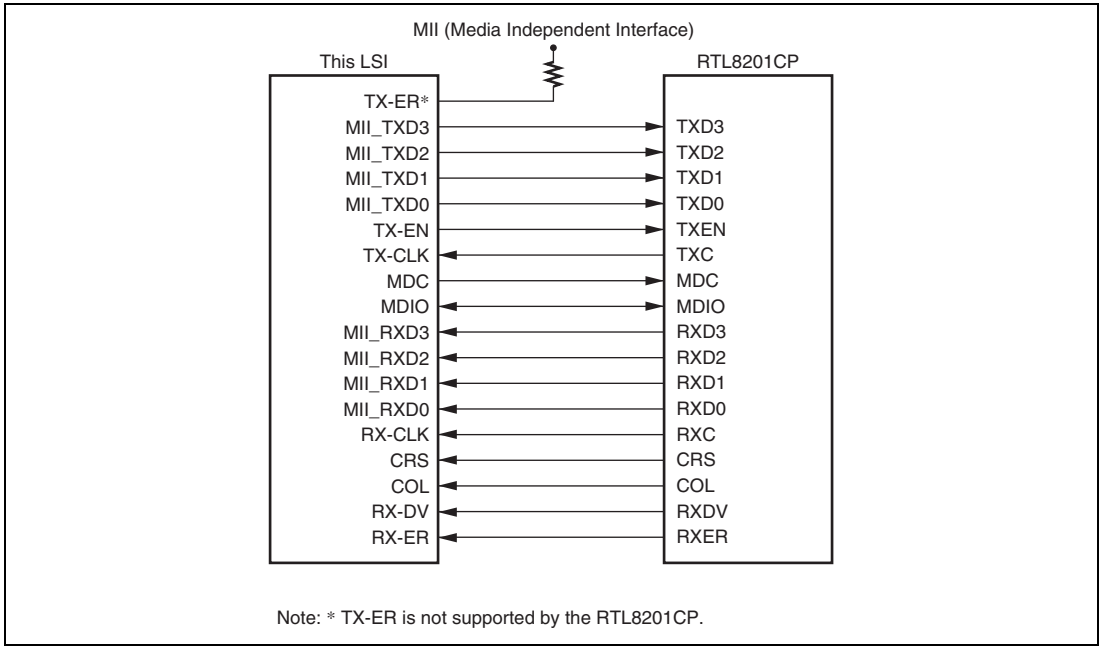


Figure 25.8 Example of Connection to RTL8201CP

25.6 Usage Notes

Pay attention to the following when using the EtherC.

(1) Conditions for setting the LCHNG bit

The LCHNG bit in ECSR may be set to 1 even when the LNKSTA pin input level remains unchanged. This may occur when the LNKSTA pin is selected by the PD19MD or PE0MD bits in the PFC or when a high level is input to the LNKSTA pin while the EtherC/E-DMAC software reset is canceled by the SWR bit in EDMR of the E-DMAC.

This is because the LNKSTA signal is internally fixed low regardless of the external pin input level when the LNKSTA pin is not selected by the PFC or while the EtherC/E-DMAC is in the software reset state.

In order not to generate a LINK signal change interrupt accidentally, clear the LCHNG bit to 0 and then set the LCHNGIP bit in ECSIPR.

To cause a transition to software standby mode, stop the EtherC/E-DMAC modules by setting the MSTP40 bit in the standby control register 4 (STBCR4) to 1.

(2) Number of Cycles for Access to Registers

Note that the number of cycles for access to EtherC registers differs from the number for access to registers in other on-chip peripheral modules (see section 9.5.12 (3), On-Chip Peripheral Module Access).

Section 26 Ethernet Controller Direct Memory Access Controller (E-DMAC) (SH7216A, SH7214A, SH7216G, and SH7214G only)

This LSI has an on-chip direct memory access controller (E-DMAC) directly connected to the Ethernet controller (EtherC). The E-DMAC controls the most part of the buffer management by using descriptors. This reduces the load on the CPU, thus enabling efficient data transmission/reception control.

Figure 26.1 shows the configuration of the E-DMAC and the descriptors and transmit/receive buffers in memory.

26.1 Features

- The load on the CPU is reduced by means of a descriptor management system.
- Transmit/receive frame status information is indicated in descriptors.
- Efficient system bus utilization is achieved through the use of DMA block transfer (32-byte units).
- Single-frame/multi-buffer operation is supported.

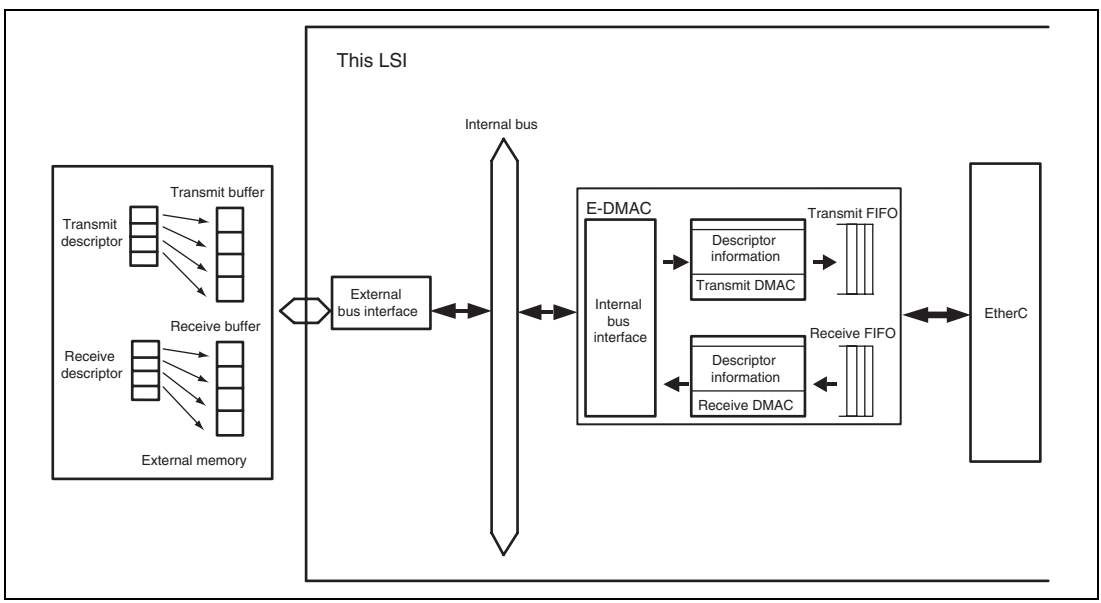


Figure 26.1 Configuration of E-DMAC, Descriptors, and Buffers

26.2 Register Descriptions

Table 26.1 shows the configuration of registers of the E-DMAC.

Table 26.1 Register Configuration

Name	Abbreviation	R/W	Address	Access Size
E-DMAC mode register	EDMR	R/W	H'FFFC 3000	32
E-DMAC transmit request register	EDTRR	R/W	H'FFFC 3008	32
E-DMAC receive request register	EDRRR	R/W	H'FFFC 3010	32
Transmit descriptor list start address register	TDLAR	R/W	H'FFFC 3018	32
Receive descriptor list start address register	RDLAR	R/W	H'FFFC 3020	32
EtherC/E-DMAC status register	EESR	R/W	H'FFFC 3028	32
EtherC/E-DMAC status interrupt enable register	EESIPR	R/W	H'FFFC 3030	32
Transmit/receive status copy enable register	TRSCER	R/W	H'FFFC 3038	32
Receive missed-frame counter register	RMFCR	R	H'FFFC 3040	32
Transmit FIFO threshold register	TFTR	R/W	H'FFFC 3048	32
FIFO depth register	FDR	R/W	H'FFFC 3050	32
Receiving method control register	RMCR	R/W	H'FFFC 3058	32
Transmit FIFO underrun counter register	TFUCR	R/W	H'FFFC 3064	32
Receive FIFO overflow counter register	RFOCR	R/W	H'FFFC 3068	32
Receive buffer write address register	RBWAR	R	H'FFFC 30C8	32
Receive descriptor fetch address register	RDFAR	R	H'FFFC 30CC	32
Transmit buffer read address register	TBRAR	R	H'FFFC 30D4	32
Transmit descriptor fetch address register	TDFAR	R	H'FFFC 30D8	32
Flow control start FIFO threshold setting register	FCFTR	R/W	H'FFFC 3070	32
Transmit interrupt setting register	TRIMD	R/W	H'FFFC 307C	32
Independent output signal setting register	IOSR	R/W	H'FFFC 306C	32
E-DMAC operation control register	EDOCR	R/W	H'FFFC 30E4	32

Table 26.2 shows the state of registers in each processing mode.

Table 26.2 Register States in Each Processing Mode

Name	Abbreviation	Software Reset
E-DMAC mode register	EDMR	Initialized
E-DMAC transmit request register	EDTRR	Initialized
E-DMAC receive request register	EDRRR	Initialized
Transmit descriptor list start address register	TDLAR	Retained
Receive descriptor list start address register	RDLAR	Retained
EtherC/E-DMAC status register	EESR	Initialized
EtherC/E-DMAC status interrupt enable register	EESIPR	Initialized
Transmit/receive status copy enable register	TRSCER	Initialized
Receive missed-frame counter register	RMFCR	Retained
Transmit FIFO threshold register	TFTR	Initialized
FIFO depth register	FDR	Initialized
Receiving method control register	RMCR	Initialized
Transmit FIFO underrun counter register	TFUCR	Retained
Receive FIFO overflow counter register	RFOCR	Retained
Receive buffer write address register	RBWAR	Initialized
Receive descriptor fetch address register	RDFAR	Initialized
Transmit buffer read address register	TBRAR	Initialized
Transmit descriptor fetch address register	TDFAR	Initialized
Flow control start FIFO threshold setting register	FCFTR	Initialized
Transmit interrupt setting register	TRIMD	Initialized
Independent output signal setting register	IOSR	Initialized

26.2.1 E-DMAC Mode Register (EDMR)

EDMR is a 32-bit readable/writable register that specifies E-DMAC operating mode. This register should usually be set at initialization after a reset. If the EtherC and E-DMAC are initialized with this register during data transmission, abnormal data may be transmitted on the line. It is prohibited to modify the operating mode while the transmission or reception function is enabled. Before changing the operating mode, the EtherC and E-DMAC should be initialized by setting the software reset bit (SWR) to 1. Note that it takes 64 cycles of internal bus clock Bϕ for the EtherC and E-DMAC to be completely initialized. Therefore, the registers in the EtherC or E-DMAC should be accessed after that.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	DE	DL[1:0]		–	–	–	SWR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6	DE	0	R/W	Big Endian/Little Endian Mode 0: Big endian (longword access) (Initial value) 1: Little endian (longword access) This setting applies to transmit and receive data, but does not apply to transmit/receive descriptors or registers (only big endian mode is available).
5, 4	DL[1:0]	00	R/W	Transmit/Receive Descriptor Length 00: 16 bytes (Initial value) 01: 32 bytes 10: 64 bytes 11: 16 bytes

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SWR	0	R/W	Software Reset [Writing] 0: Disabled 1: Internal hardware is reset. For the registers that are reset, see tables 25.3 and 26.2.

26.2.2 E-DMAC Transmit Request Register (EDTRR)

EDTRR is a 32-bit readable/writable register that issues transmit directives to the E-DMAC. After having transmitted one frame, the E-DMAC reads the next descriptor. When the TACT bit in this descriptor is set to 1 (valid), the E-DMAC continues transmission. Otherwise, the E-DMAC clears the TR bit and stops the transmit DMAC operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	TR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TR	0	R/W	Transmit Request 0: Transmission-halted state. Writing 0 does not stop transmission. Termination of transmission is controlled by the TACT bit of the transmit descriptor. 1: Transmission start. The relevant descriptor is read and the frame in which the TACT bit is 1 is transmitted.

26.2.3 E-DMAC Receive Request Register (EDRRR)

EDRRR is a 32-bit readable/writable register that issues receive directives to the E-DMAC. After writing 1 to the RR bit in this register, the E-DMAC reads the receive descriptor. When the RACT bit in this receive descriptor is set to 1 (valid), the E-DMAC prepares for a receive request from the EtherC. When reception of data for the receive buffer is completed, the E-DMAC reads the next receive descriptor and prepares for receiving frames. If the RACT bit in the receive descriptor is cleared to 0 (invalid) at this time, the E-DMAC clears the RR bit and stops the receive DMAC operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	RR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RR	0	R/W	Receive Request 0: Receiving function is disabled* 1: Receive descriptor is read, and the E-DMAC is ready to receive

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 E-DMAC cannot operate successfully. In this case, to make E-DMAC reception enabled again, execute a software reset by the SWR bit in EDMR.

To disable the E-DMAC receiving function without executing a software reset, clear the RE bit in ECMR of the EtherC to 0. Next, after the E-DMAC has completed the reception and write-back to the receive descriptor has been confirmed, disable the receiving function using this register.

26.2.4 Transmit Descriptor List Start Address Register (TDLAR)

TDLAR is a 32-bit readable/writable register that specifies the start address of the transmit descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during transmission, and must be modified while the TR bit in the E-DMAC transmit request register (EDTRR) is 0 (transmission-halted state).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDLA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDLA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDLA[31:0]	All 0	R/W	Transmit Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: TDLA[3:0] = 0000 32-byte boundary: TDLA[4:0] = 00000 64-byte boundary: TDLA[5:0] = 000000

26.2.5 Receive Descriptor List Start Address Register (RDLAR)

RDLAR is a 32-bit readable/writable register that specifies the start address of the receive descriptor list. Descriptors have a boundary configuration in accordance with the descriptor length indicated by the DL bits in EDMR. This register must not be modified during reception, and must be modified while the RR bit in the E-DMAC receive request register (EDRRR) is 0 (reception-disabled state).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDLA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDLA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDLA[31:0]	All 0	R/W	Receive Descriptor Start Address The lower bits are set according to the specified descriptor length. 16-byte boundary: RDLA[3:0] = 0000 32-byte boundary: RDLA[4:0] = 00000 64-byte boundary: RDLA[5:0] = 000000

26.2.6 EtherC/E-DMAC Status Register (EESR)

EESR is a 32-bit readable/writable register that indicates communications status information on the E-DMAC in combination with the EtherC. The information in this register is reported in the form of interrupt sources. Individual bits are cleared by writing 1 (except for read-only bit 22 (ECI)), and are not affected by writing 0. Each interrupt source can be masked by the corresponding bit in the EtherC/E-DMAC status interrupt enable register (EESIPR).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	TWB	–	–	–	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	CND	DLC	CD	TRO	RMAF	–	–	RRF	RTLF	RTSF	PRE	CERF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	TWB	0	R/W	Write-Back Completed Indicates that write-back from the E-DMAC to the corresponding descriptor after frame transmission has been completed. This operation is enabled only when the TIS bit in TRIMD is set to 1. 0: Write-back has not been completed or no transmission directive is given 1: Write-back has been completed
29 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26	TABT	0	R/W	Transmit Abort Detect Indicates that the EtherC has aborted sending a frame because of an error or fault during frame transmission. 0: Frame transmission has not been aborted or no transmission directive is given 1: Frame transmission has been aborted

Bit	Bit Name	Initial Value	R/W	Description
25	RABT	0	R/W	Receive Abort Detect Indicates that the EtherC has aborted receiving a frame because of an error or fault during frame reception. 0: Frame reception has not been aborted or no reception directive is given 1: Frame reception has been aborted
24	RFCOF	0	R/W	Receive Frame Counter Overflow Indicates that the frame counter in the receive FIFO has overflowed. 0: Receive frame counter has not overflowed 1: Receive frame counter has overflowed
23	ADE	0	R/W	Address Error Indicates that the memory address that the E-DMAC tried to transfer is found incorrect. 0: Incorrect memory address has not been detected (normal operation) 1: Incorrect memory address has been detected Note: When an address error is detected, the E-DMAC stops transmitting/receiving data. To resume the operation, execute a software reset with the SWR bit in EDMR.
22	ECI	0	R	EtherC Status Register Source This bit is a read-only bit. When the source of an ECSR interrupt is cleared, this bit is also cleared. 0: EtherC status interrupt source has not been detected 1: EtherC status interrupt source has been detected

Bit	Bit Name	Initial Value	R/W	Description
21	TC	0	R/W	Frame Transmit Completed Indicates that all the data specified by the transmit descriptor has been transmitted from the EtherC. This bit is set to 1, assuming the completion of transmission, when transmission of one frame is completed in the single-frame/single-buffer processing or when the last data of a frame has been transmitted and the transmit descriptor active bit (TACT) of the next descriptor is not set in the multi-buffer frame processing. After frame transmission, the E-DMAC writes the transfer status back to the relevant descriptor. 0: Transfer is not completed or no transfer directive is given 1: Transfer is completed
20	TDE	0	R/W	Transmit Descriptor Empty Indicates that the transmit descriptor active bit (TACT) in a transmit descriptor is not set when it is read by the E-DMAC if the previous descriptor does not represent the end of a frame in the multi-buffer frame processing. As a result, an incomplete frame may be sent. 0: Transmit descriptor active bit TACT = 1 detected 1: Transmit descriptor active bit TACT = 0 detected 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 the transmit descriptor list start address register (TDLAR).
19	TFUF	0	R/W	Transmit FIFO Underflow Indicates that an underflow has occurred in the transmit FIFO during frame transmission. Incomplete data is sent onto the line. 0: Underflow has not occurred 1: Underflow has occurred

Bit	Bit Name	Initial Value	R/W	Description
18	FR	0	R/W	Frame Reception Indicates that a frame has been received and the receive descriptor has been updated. This bit is set to 1 each time a frame is received. 0: Frame has not been received 1: Frame has been received
17	RDE	0	R/W	Receive Descriptor Empty When receive descriptor empty (RDE = 1) occurs, reception can be resumed by setting the RACT bit in the receive descriptor to 1 to restart the receive operation. 0: Receive descriptor active bit RACT = 1 detected 1: Receive descriptor active bit RACT = 0 detected
16	RFOF	0	R/W	Receive FIFO Overflow Indicates that the receive FIFO has overflowed during frame reception. 0: Overflow has not occurred 1: Overflow has occurred
15 to 12	—	All 0	R	Reserved The write value should always be 0.
11	CND	0	R/W	Carrier Not Detect Indicates the carrier detection status. 0: Carrier has been detected when transmission starts 1: Carrier has not been detected
10	DLC	0	R/W	Carrier Loss Detect Indicates that loss of carrier has been detected during frame transmission. 0: Loss of carrier has not been detected 1: Loss of carrier has been detected
9	CD	0	R/W	Delayed Collision Detect Indicates that a delayed collision has been detected during frame transmission. 0: Delayed collision has not been detected 1: Delayed collision has been detected

Bit	Bit Name	Initial Value	R/W	Description
8	TRO	0	R/W	Transmit Retry Limit Exceeded Indicates that a retry limit exceeded condition has occurred during frame transmission. Total 16 transmission retries including 15 retransmission attempts based on the back-off algorithm have failed after the EtherC started transmission. 0: Transmit retry limit exceeded condition has not been detected 1: Transmit retry limit exceeded condition has been detected
7	RMAF	0	R/W	Receive Multicast Address Frame 0: Multicast address frame has not been received 1: Multicast address frame has been received
6, 5	—	All 0	R	Reserved The write value should always be 0.
4	RRF	0	R/W	Receive Residual-Bit Frame 0: Residual-bit frame has not been received 1: Residual-bit frame has been received
3	RTLF	0	R/W	Receive Too-Long Frame Indicates that a frame longer than the receive frame length upper limit set by RFLR in EtherC has been received. 0: Too-long frame has not been received 1: Too-long frame has been received
2	RTSF	0	R/W	Receive Too-Short Frame Indicates that a frame shorter than 64 bytes has been received. 0: Too-short frame has not been received 1: Too-short frame has been received
1	PRE	0	R/W	PHY-LSI Receive Error 0: PHY-LSI receive error has not been detected 1: PHY-LSI receive error has been detected

Bit	Bit Name	Initial Value	R/W	Description
0	CERF	0	R/W	Receive Frame CRC Error 0: CRC error has not been detected 1: CRC error has been detected

26.2.7 EtherC/E-DMAC Status Interrupt Enable Register (EESIPR)

EESIPR is a 32-bit readable/writable register that enables interrupts corresponding to individual bits in the EtherC/E-DMAC status register (EESR). An interrupt is enabled by writing 1 to the corresponding bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	TWB IP	–	–	–	TABT IP	RABT IP	RFCOF IP	ADE IP	ECI IP	TC IP	TDE IP	TFUF IP	FR IP	RDE IP	RFOF IP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	CND IP	DLC IP	CD IP	TRO IP	RMAF IP	–	–	RRF IP	RTLF IP	RTSF IP	PRE IP	CERF IP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	TWBIP	0	R/W	Write-Back Complete Interrupt Enable 0: Write-back complete interrupt is disabled 1: Write-back complete interrupt is enabled
29 to 27	—	All 0	R	Reserved The write value should always be 0.
26	TABTIP	0	R/W	Transmit Abort Detect Interrupt Enable 0: Transmit abort detect interrupt is disabled 1: Transmit abort detect interrupt is enabled
25	RABTIP	0	R/W	Receive Abort Detect Interrupt Enable 0: Receive abort detect interrupt is disabled 1: Receive abort detect interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
24	RFCOFIP	0	R/W	Receive Frame Counter Overflow Interrupt Enable 0: Receive frame counter overflow interrupt is disabled 1: Receive frame counter overflow interrupt is enabled
23	ADEIP	0	R/W	Address Error Interrupt Enable 0: Address error interrupt is disabled 1: Address error interrupt is enabled
22	ECIIP	0	R/W	EtherC Status Register Source Interrupt Enable 0: EtherC status interrupt is disabled 1: EtherC status interrupt is enabled
21	TCIP	0	R/W	Frame Transmission Complete Interrupt Enable 0: Frame transmission complete interrupt is disabled 1: Frame transmission complete interrupt is enabled
20	TDEIP	0	R/W	Transmit Descriptor Empty Interrupt Enable 0: Transmit descriptor empty interrupt is disabled 1: Transmit descriptor empty interrupt is enabled
19	TFUFIP	0	R/W	Transmit FIFO Underflow Interrupt Enable 0: Underflow interrupt is disabled 1: Underflow interrupt is enabled
18	FRIP	0	R/W	Frame Reception Interrupt Enable 0: Frame reception interrupt is disabled 1: Frame reception interrupt is enabled
17	RDEIP	0	R/W	Receive Descriptor Empty Interrupt Enable 0: Receive descriptor empty interrupt is disabled 1: Receive descriptor empty interrupt is enabled
16	RFOFIP	0	R/W	Receive FIFO Overflow Interrupt Enable 0: Overflow interrupt is disabled 1: Overflow interrupt is enabled
15 to 12	—	All 0	R	Reserved The write value should always be 0.
11	CNDIP	0	R/W	Carrier Not Detect Interrupt Enable 0: Carrier not detect interrupt is disabled 1: Carrier not detect interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
10	DLCIP	0	R/W	Carrier Loss Detect Interrupt Enable 0: Carrier loss detect interrupt is disabled 1: Carrier loss detect interrupt is enabled
9	CDIP	0	R/W	Delayed Collision Detect Interrupt Enable 0: Delayed collision detect interrupt is disabled 1: Delayed collision detect interrupt is enabled
8	TROIP	0	R/W	Transmit Retry Over Interrupt Enable 0: Transmit retry over interrupt is disabled 1: Transmit retry over interrupt is enabled
7	RMAFIP	0	R/W	Multicast Address Frame Reception Interrupt Enable 0:Multicast address frame reception interrupt is disabled 1: Multicast address frame reception interrupt is enabled
6, 5	—	All 0	R	Reserved The write value should always be 0.
4	RRFIP	0	R/W	Residual-Bit Frame Reception Interrupt Enable 0: Residual-bit frame reception interrupt is disabled 1: Residual-bit frame reception interrupt is enabled
3	RTLFIP	0	R/W	Too-Long Frame Reception Interrupt Enable 0: Too-long frame reception interrupt is disabled 1: Too-long frame reception interrupt is enabled
2	RTSFIP	0	R/W	Too-Short Frame Reception Interrupt Enable 0: Too-short frame reception interrupt is disabled 1: Too-short frame reception interrupt is enabled
1	PREIP	0	R/W	PHY-LSI Receive Error Interrupt Enable 0: PHY-LSI receive error interrupt is disabled 1: PHY-LSI receive error interrupt is enabled
0	CERFIP	0	R/W	Receive Frame CRC Error Interrupt Enable 0: CRC error interrupt is disabled 1: CRC error interrupt is enabled

26.2.8 Transmit/Receive Status Copy Enable Register (TRSCER)

TRSCER specifies whether to reflect the transmit/receive status information reported by bits in the EtherC/E-DMAC status register (EESR) in bits TFS25 to TFS0 or RFS26 to RFS0 of the corresponding descriptor. The bits in this register 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.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	CNDCE	DLCCE	CDCE	TROCE	RMAFCE	–	–	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	CNDCE	0	R/W	CND Bit Copy Directive 0: Reflects the CND bit status in the TFS bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFS bit of the transmit descriptor
10	DLCCE	0	R/W	DLC Bit Copy Directive 0: Reflects the DLC bit status in the TFS bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFS bit of the transmit descriptor
9	CDCE	0	R/W	CD Bit Copy Directive 0: Reflects the CD bit status in the TFS bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFS bit of the transmit descriptor

Bit	Bit Name	Initial Value	R/W	Description
8	TROCE	0	R/W	TRO Bit Copy Directive 0: Reflects the TRO bit status in the TFS bit of the transmit descriptor 1: Occurrence of the corresponding source is not reflected in the TFS bit of the transmit descriptor
7	RMAFCE	0	R/W	RMAF Bit Copy Directive 0: Reflects the RMAF bit status in the RFS bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFS bit of the receive descriptor
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	RRFCE	0	R/W	RRF Bit Copy Directive 0: Reflects the RRF bit status in the RFS bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFS bit of the receive descriptor
3	RTLFCE	0	R/W	RTLF Bit Copy Directive 0: Reflects the RTLF bit status in the RFS bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFS bit of the receive descriptor
2	RTSFCE	0	R/W	RTSF Bit Copy Directive 0: Reflects the RTSF bit status in the RFS bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFS bit of the receive descriptor

Bit	Bit Name	Initial Value	R/W	Description
1	PRECE	0	R/W	PRE Bit Copy Directive 0: Reflects the PRE bit status in the RFS bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFS bit of the receive descriptor
0	CERFCE	0	R/W	CERF Bit Copy Directive 0: Reflects the CERF bit status in the RFS bit of the receive descriptor 1: Occurrence of the corresponding source is not reflected in the RFS bit of the receive descriptor

26.2.9 Receive Missed-Frame Counter Register (RMFCR)

RMFCR is a 16-bit counter that indicates the number of frames that were not saved 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 this register reaches H'FFFF, the counter stops incrementing. The counter value is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MFC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	MFC[15:0]	All 0	R	Missed-Frame Counter These bits indicate the number of frames that were not transferred to the receive buffer and were discarded during reception.

26.2.10 Transmit FIFO Threshold Register (TFTR)

TFTR is a 32-bit readable/writable register that specifies the transmit FIFO threshold at which the first transmission is started. The actual threshold is 4 times the set value. The EtherC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes specified by this register, when the transmit FIFO is full, or when one-frame data is written. Set this register in the transmission-halted state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	TFT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	TFT[10:0]	All 0	R/W	Transmit FIFO Threshold A value smaller than the FIFO size specified by FDR must be set as the transmit FIFO threshold. H'000: Store and forward mode H'001 to H'00C: Setting prohibited H'00D: 52 bytes H'00E: 56 bytes : : H'01F: 124 bytes H'020: 128 bytes : : H'03F: 252 bytes H'040: 256 bytes : : H'07F: 508 bytes H'080: 512 bytes : : H'0FF: 1,020 bytes H'100: 1,024 bytes : : H'1FF: 2,044 bytes H'200: 2,048 bytes H'201 to H'7FF: Setting prohibited

Notes: 1. When starting transmission before one-frame data write has been completed, take care no underflow occurs.
2. Operation cannot be guaranteed when the value of this register is greater than the transmit FIFO or receive FIFO size.
3. To prevent a transmit underflow, setting the initial value (store and forward mode) is recommended.

26.2.11 FIFO Depth Register (FDR)

FDR is a 32-bit readable/writable register that specifies the sizes of the transmit FIFO and receive FIFO.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	TFD[4:0]					–	–	–	RFD[4:0]				
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 to 8	TFD[4:0]	00111	R/W	Transmit FIFO Size Specifies the size of the transmit FIFO. The setting must not be changed during transmission or reception. 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 above: Setting prohibited
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4 to 0	RFD[4:0]	00111	R/W	Receive FIFO Size Specifies the size of the receive FIFO. The setting must not be changed during transmission or reception. 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 above: Setting prohibited

Note: Operation cannot be guaranteed when the value set in this register is greater than the transmit FIFO or receive FIFO size.

26.2.12 Receiving Method Control Register (RMCR)

RMCR is a 32-bit readable/writable register that specifies how to control the RR bit in EDRRR when a frame is received. Set this register in the reception idle state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	RNC	RNR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	RNC	0	R/W	Receive Request Bit Non-Reset Mode 0: No operation 1: Allows the software to reset the receive request (RR) bit in EDRRR. Even when the RACT bit in the fetched descriptor is 0 (receive descriptor empty), the receive request bit (RR) in EDRRR is not automatically reset and the receive descriptor is continuously fetched to continue DMA transfer of receive frames.

Bit	Bit Name	Initial Value	R/W	Description
0	RNR	0	R/W	Receive Request Bit Reset 0: Allows the hardware to reset the receive request (RR) bit in EDRRR automatically upon completion of reception of one frame. This control is possible for each frame. To receive the subsequent receive frame, the RR bit in EDRRR needs to be set again. 1: Allows the higher-level software to control the receive request (RR) bit in EDRRR. Once the RR bit in EDRRR is set to 1, the hardware continues to fetch the receive descriptor and receive frames autonomously until the RR bit in EDRRR is cleared to 0. In other words, continuous reception of multiple frames are possible. Setting this bit to 1 is recommended for continuous reception. However, when a receive descriptor empty is detected, the hardware clears the RR bit in EDRRR automatically.

26.2.13 Transmit FIFO Underrun Counter Register (TFUCR)

TFUCR is a register that indicates the number of underruns having occurred in the transmit FIFO. The counter is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UNDER[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	UNDER[15:0]	All 0	R/W	Transmit FIFO Underflow Count Indicates the count of underflows having occurred in the transmit FIFO. The counter stops when the count value reaches H'FFFF.

26.2.14 Receive FIFO Overflow Counter Register (RFOCR)

RFOCR is a register that indicates the number of overflows having occurred in the receive FIFO. The counter is cleared to 0 by writing any value to this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVER[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	OVER[15:0]	All 0	R/W	Receive FIFO Overflow Count Indicates the count of overflows having occurred in the receive FIFO. The counter stops when the count value reaches H'FFFF.

26.2.15 Receive Buffer Write Address Register (RBWAR)

RBWAR stores the buffer address of data to be written in the receive buffer when the E-DMAC writes data to the receive buffer. Which addresses in the receive buffer are processed by the E-DMAC can be recognized by monitoring the address specified in this register. The address that the E-DMAC is actually accessing during the buffer write processing is not always equal to the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBWA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBWA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RBWA[31:0]	All 0	R	Receive Buffer Write Address These bits can only be read. Writing is prohibited.

26.2.16 Receive Descriptor Fetch Address Register (RDFAR)

RDFAR stores the descriptor start address required when the E-DMAC fetches descriptor information from the receive descriptor. Which receive descriptor information is used for processing by the E-DMAC can be recognized by monitoring the addresses indicated by this register. The address that the E-DMAC is actually accessing during the descriptor fetch processing is not always equal to the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDFA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDFA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDFA[31:0]	All 0	R	Receive Descriptor Fetch Address These bits can only be read. Writing is prohibited.

26.2.17 Transmit Buffer Read Address Register (TBRAR)

TBRAR stores the address of the transmit buffer from which the E-DMAC reads data. Which address in the transmit buffer is being processed by the E-DMAC can be recognized by monitoring the address indicated by this register. The address that the E-DMAC is actually accessing during the buffer read processing is not always equal to the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBRA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TBRA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBRA[31:0]	All 0	R	Transmit Buffer Read Address These bits can only be read. Writing is prohibited.

26.2.18 Transmit Descriptor Fetch Address Register (TDFAR)

TDFAR stores the descriptor start address that is required when the E-DMAC fetches descriptor information from the transmit descriptor. Which transmit descriptor information is used for processing by the E-DMAC can be recognized by monitoring the address indicated by this register. The address that the E-DMAC is actually accessing during the descriptor fetch processing is not always equal to the value read from this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDFA[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDFA[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TDFA[31:0]	All 0	R	Transmit Descriptor Fetch Address These bits can only be read. Writing is prohibited.

26.2.19 Flow Control Start FIFO Threshold Setting Register (FCFTR)

FCFTR is a 32-bit readable/writable register that sets the flow control of the EtherC (automatic PAUSE transmission threshold setting). FCFTR can set the threshold values for the receive FIFO data size (RFDO[2:0]) and the number of receive frames (RFFO[2:0]). The flow control starts when either the receive FIFO data size threshold or the receive frame count threshold is determined.

If the same receive FIFO size as set by the FIFO depth register (FDR) is set when the flow control is to be turned on according to the RFDO setting condition, flow control is turned on with (FIFO data size − 64) bytes. When RFD = 00111 in FDR and RFDO = 111 in this register, for instance, the flow control is turned on when (2,048 − 64) bytes of data are stored in the receive FIFO. Set a value equal to or less than the RFD value in FDR for the RFDO bits in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	RFFO[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	RFDO[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	RFFO[2:0]	111	R/W	Receive Frame Count Overflow BSY Output Threshold 000: When two frames have been stored in the receive FIFO. 001: When four frames have been stored in the receive FIFO. 010: When six frames have been stored in the receive FIFO. : 110: When 14 frames have been stored in the receive FIFO. 111: When 16 frames have been stored in the receive FIFO.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	RFDO[2:0]	111	R/W	Receive FIFO Overflow BSY Output Threshold 000: When (256 – 32)-byte data is stored in the receive FIFO. 001: When (512 – 32)-byte data is stored in the receive FIFO. : 110: When (1792 – 32)-byte data is stored in the receive FIFO. 111: When (2048 – 64)-byte data is stored in the receive FIFO.

26.2.20 Transmit Interrupt Setting Register (TRIMD)

TRIMD is a 32-bit readable/writable register that specifies whether to notify write-back completion of each frame during transmission with the TWB bit in EESR or an interrupt.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	TIM	–	–	–	TIS
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TIM	0	R/W	Transmit Interrupt Mode 0: Per-transmit-frame mode An interrupt is notified upon completion of write-back of each transmit frame. 1: Interrupt mode An interrupt is notified upon completion of write-back to the transmit descriptor with the TWBI bit set to 1.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TIS	0	R/W	Transmit Interrupt Setting 0: Interrupt not set An interrupt is not notified in the mode specified by the TIM bit. When this bit is 0, the TIM bit setting is invalid. 1: Interrupt set An interrupt is notified by setting the TWB bit in EESR to 1 in the mode specified by the TIM bit.

26.2.21 Independent Output Signal Setting Register (IOSR)

The ELB bit value in this register is directly output to the general external output pin (EXOUT) of this LSI. The EXOUT pin can be used to specify loopback mode for the PHY-LSI. To achieve the loopback function for the PHY-LSI with this register, the PHY-LSI must have a pin corresponding to the EXOUT pin.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ELB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ELB	0	R/W	External Loopback Mode 0: The EXOUT pin outputs a low level signal. 1: The EXOUT pin outputs a high level signal.

26.2.22 E-DMAC Operation Control Register (EDOCR)

EDOCR is a 32-bit readable/writable register that specifies control methods in each operation status of the E-DMAC.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	FEC	AEC	EDH	NMIE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/(W)*	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	FEC	0	R/W	FIFO Error Control Specifies the E-DMAC operation when a transmit FIFO underflow or a receive FIFO overflow occurs. 0: The E-DMAC continues operating even when an underflow or overflow occurs 1: The E-DMAC stops operating when an underflow or overflow occurs
2	AEC	0	R/W	Address Error Detect Indicates that the memory address that the E-DMAC is going to transfer is incorrect. 0: Incorrect memory address has not been detected (normal operation) 1: The E-DMAC stops operating due to incorrect memory address Note: To resume the E-DMAC operation, issue a software reset with the SWR bit in EDMR and then make settings again.

Bit	Bit Name	Initial Value	R/W	Description
1	EDH	0	R/(W)*	NMI Interrupt Detect 0: No NMI interrupt has been detected 1: An NMI interrupt has been detected The E-DMAC stops operating when an NMI interrupt is detected while NMIE = 0. Note: Only writing 0 after reading 1 is enabled.
0	NMIE	0	R/W	NMI Interrupt Control 0: The E-DMAC stops operating when an NMI interrupt is detected 1: The E-DMAC continues to operate even when an NMI interrupt is detected

26.3 Operation

The E-DMAC, connected to the EtherC, allows efficient transfer of transmit/receive data between the EtherC and memory (buffers) without CPU intervention. The E-DMAC automatically reads the control information referred to as descriptors. The descriptors corresponding to each buffer store buffer pointers and other information. The E-DMAC reads transmit data from the transmit buffer and writes receive data to the receive buffer according to the control information. Arranging such multiple descriptors in a row (i.e., making a descriptor list) allows continuous transmission or reception.

26.3.1 Descriptor Lists and Data Buffers

The communication program creates a transmit descriptor list and a receive descriptor list in a memory space prior to transmission and reception, and sets the start addresses of these lists to 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 by the E-DMAC mode register (EDMR). The start address of the transmit buffer can be placed on a longword, word, or byte boundary.

(1) Transmit Descriptor

Figure 26.2 shows the relationship between a transmit descriptor and a transmit buffer. The descriptor can relate one transmit frame to one transmit buffer (single-frame/single-buffer operation) or 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 below 42 bytes, operation cannot be guaranteed.

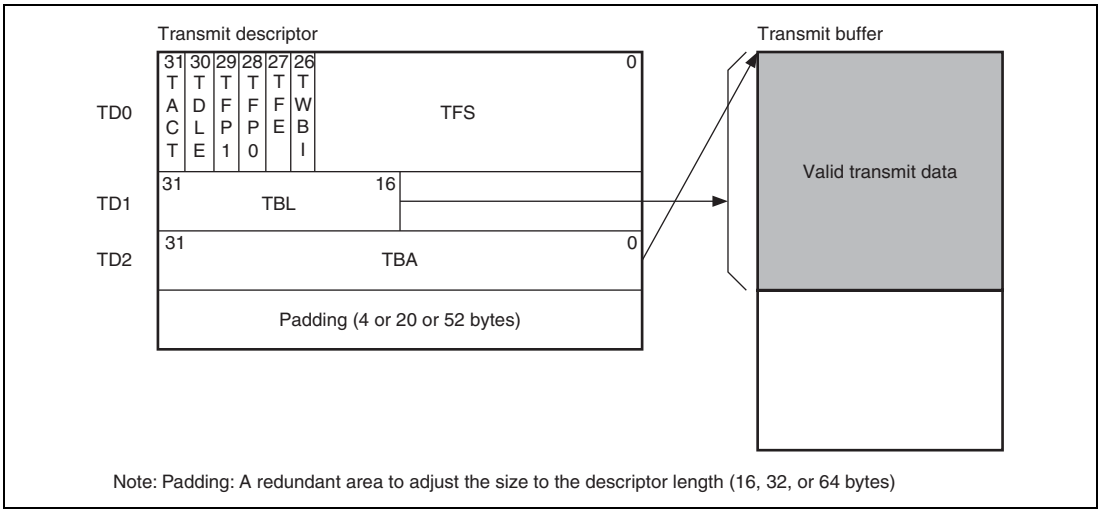


Figure 26.2 Relationship between Transmit Descriptor and Transmit Buffer

(a) Transmit Descriptor 0 (TD0)

TD0 indicates the transmit frame status informing frame transmission status.

(The underlined bits in the table below are subject to write-back.)

Bit	Bit Name	Initial Value	R/W	Description
31	TACT	0	R/W	Transmit Descriptor Valid Indicates that the corresponding descriptor is valid. This bit is set to 1 by software, and is cleared to 0 by hardware when a transmit frame has been completely transferred or when transmission has been aborted due to some cause.
30	TDLE	0	R/W	Transmit Descriptor Ring End When set to 1, this bit indicates that the corresponding descriptor is the last one of the transmit descriptor ring.
29	TFP1	0	R/W	Transmit Frame Positions 1 and 0
28	TFP0	0	R/W	These bits relate the transmit buffer to the transmit frame. The settings of these bits and the TBL bits should be in a logically correct relation in the consecutive descriptors. 00: Transmission of the frame of the transmit buffer specified by this descriptor is continued. (The frame is incomplete.) 01: The transmit buffer specified by this descriptor contains the end of the frame. (The frame is complete.) 10: The transmit buffer specified by this descriptor is the start of the frame. (The frame is incomplete.) 11: The content of the transmit buffer specified by this descriptor corresponds to one frame (single-frame/single-buffer).
27	TFE	0	R/W	Transmit Frame Error When set to 1, this bit indicates that an error is indicated by any of the TFS bits. (For TFS7 to TFS0, it is possible to prevent this bit from being set by TRSCER. This is not possible, however, if a source indicated by TFS7 to TFS0 also causes TFS8 to be set.) 1: Frame transmission has been aborted.

Bit	Bit Name	Initial Value	R/W	Description
26	TWBI	0	R/W	Write-Back Completion Interrupt Notification (This bit is valid when TRIMD is set so.) 0: No operation 1: An interrupt is generated upon completion of write-back to this descriptor.
25 to 0	TFS	All 0	R/W	Transmit Frame Status TFS25 toTFS9 [Reserved (The write value should always be 0.)] TFS8 [Transmit Abort Detect]: 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 [No Carrier Detect (corresponding to the CND bit in EESR)] TFS2 [Carrier Loss Detect (corresponding to the DLC bit in EESR)] TFS1 [Delayed Collision Detect during Transmission (corresponding to the CD bit in EESR)] TFS0 [Transmit Retry Over (corresponding to the TRO bit in EESR)]: 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.)

(b) Transmit Descriptor 1 (TD1)

TD1 indicates the length of the transmit buffer.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	TBL	All 0	R/W	Transmit Buffer Length Indicates the length of valid bytes of the relevant transmit buffer.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

(c) Transmit Descriptor 2 (TD2)

TD2 indicates the start address of the relevant transmit buffer.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TBA	All 0	R/W	Transmit Buffer Address Indicates the start address of the transmit buffer.

(2) Receive Descriptor

Figure 26.3 shows the relationship between a receive descriptor and a receive buffer. The receive buffer address should be set on a 32-byte boundary.

When the receive buffer length (RBL) is set to 0 byte, operation specified by the descriptor cannot be guaranteed.

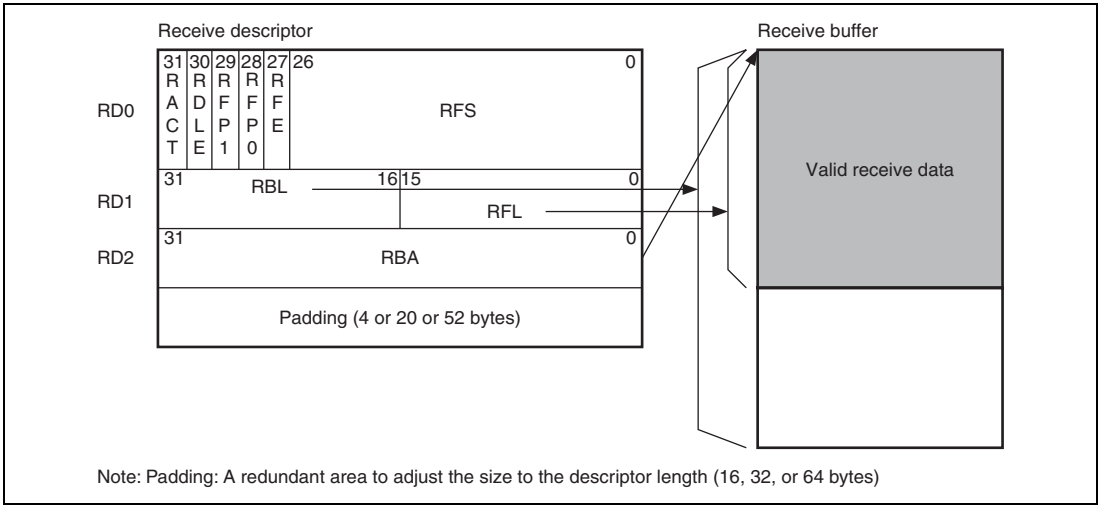


Figure 26.3 Relationship between Receive Descriptor and Receive Buffer

(a) Receive Descriptor 0 (RD0)

RD0 indicates the receive frame status informing frame reception status.

(The underlined bits in the table below are subject to write-back.)

Bit	Bit Name	Initial Value	R/W	Description
31	RACT	0	R/W	Receive Descriptor Valid Indicates that the corresponding descriptor is valid. This bit is set to 1 by software, and is cleared to 0 by hardware when a receive frame has been completely transferred to the buffer address specified by RD2 or when the receive buffer becomes full.
30	RDLE	0	R/W	Receive Descriptor Ring End When set to 1, this bit indicates that the corresponding descriptor is the last one of the receive descriptor ring.
29, 28	RFP[1:0]	00	R/W	Receive Frame Positions 1 and 0 These bits relate the receive buffer to the receive frame. 00: Reception of the frame of the receive buffer specified by this descriptor is continued. (The frame is incomplete.) 01: The receive buffer specified by this descriptor contains the end of the frame. (The frame is complete.) 10: The receive buffer specified by this descriptor is the start of the frame. (The frame is incomplete.) 11: The content of the receive buffer specified by this descriptor corresponds to one frame (single-frame/single-buffer).
27	RFE	0	R/W	Receive Frame Error When set to 1, this bit indicates that an error is indicated by any of the RFS bits. (For RFS7 to RFS0, it is possible to prevent this bit from being set by TRSCER. This is not possible, however, if a source indicated by RFS7 to RFS0 also causes RFS8 to be set.)

Bit	Bit Name	Initial Value	R/W	Description
26 to 0	RFS	All 0	R/W	Receive Frame Status RF26 to RF10 [Reserved (The write value should always be 0.)] RFS9 [Receive FIFO Overflow (corresponding to the RFOF bit in EESR)]: 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 (causing RFE to be set). RFS8 [Receive Abort Detect]: When set to 1, this bit indicates that the abort signal is set to 1 during frame reception (causing RFE to be set). RFS7 [Multicast Address Frame Received (corresponding to the RMAF bit in EESR)] RFS6 and RFS5 [Reserved (The write value should always be 0.)] RFS4 [Residual-Bit Frame Receive Error (corresponding to the RRF bit in EESR)] RFS3 [Long Frame Receive Error (corresponding to the RTLF bit in EESR)] RFS2 [Short Frame Receive Error (corresponding to the RTSF bit in EESR)] RFS1 [PHY-LSI Receive Error (corresponding to the PRE bit in EESR)] RFS0 [Receive Frame CRC Error Detect (corresponding to the CERF bit in EESR)]: When set to 1, these bits indicate that RFS8 to RFS1 have been set to 1 during frame reception. (Although RFE is normally set when these bits are set to 1, it can be prevented from being set by so setting TRSCER.)

(b) Receive Descriptor 1 (RD1)

RD1 indicates the length of the receive buffer.

(The underlined bits in the table below are subject to write-back.)

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	RBL	All 0	R/W	Receive Buffer Length Indicates the length of bytes of the relevant receive buffer. A multiple of 32 should be set for the buffer length.
15 to 0	RFL	All 0	R/W	Receive Frame Length Indicates the length (number of bytes) of a receive frame stored in the buffer.

(c) Receive Descriptor 2 (RD2)

RD2 indicates the start address of the relevant receive buffer.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RBA	All 0	R/W	Receive Buffer Address Indicates the start address of the receive buffer. The buffer address should be set on a 32-byte boundary.

26.3.2 Transmission

When the transmit request bit (TR) in the E-DMAC transmit request register (EDTRR) is set while the transmission function is enabled, the E-DMAC 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) in the initial state). When the TACT bit of the read descriptor is set to 1 (valid), the E-DMAC reads transmit frame data sequentially 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 TFP value.

- TFP = 00 or 10 (frame continuation):
  Descriptor write-back (TACT bit only) is performed after DMA transfer.
- TFP = 01 or 11 (frame end):
  Descriptor write-back (TACT bit and status) is performed upon completion of frame transmission.

As long as the TACT bit of a read descriptor is set to 1 (valid), the reading of E-DMAC descriptors and the transmission of frames continue. When a descriptor with the TACT bit cleared to 0 (invalid) is read, the E-DMAC clears the TR bit in EDTRR to 0 and completes transmit processing.

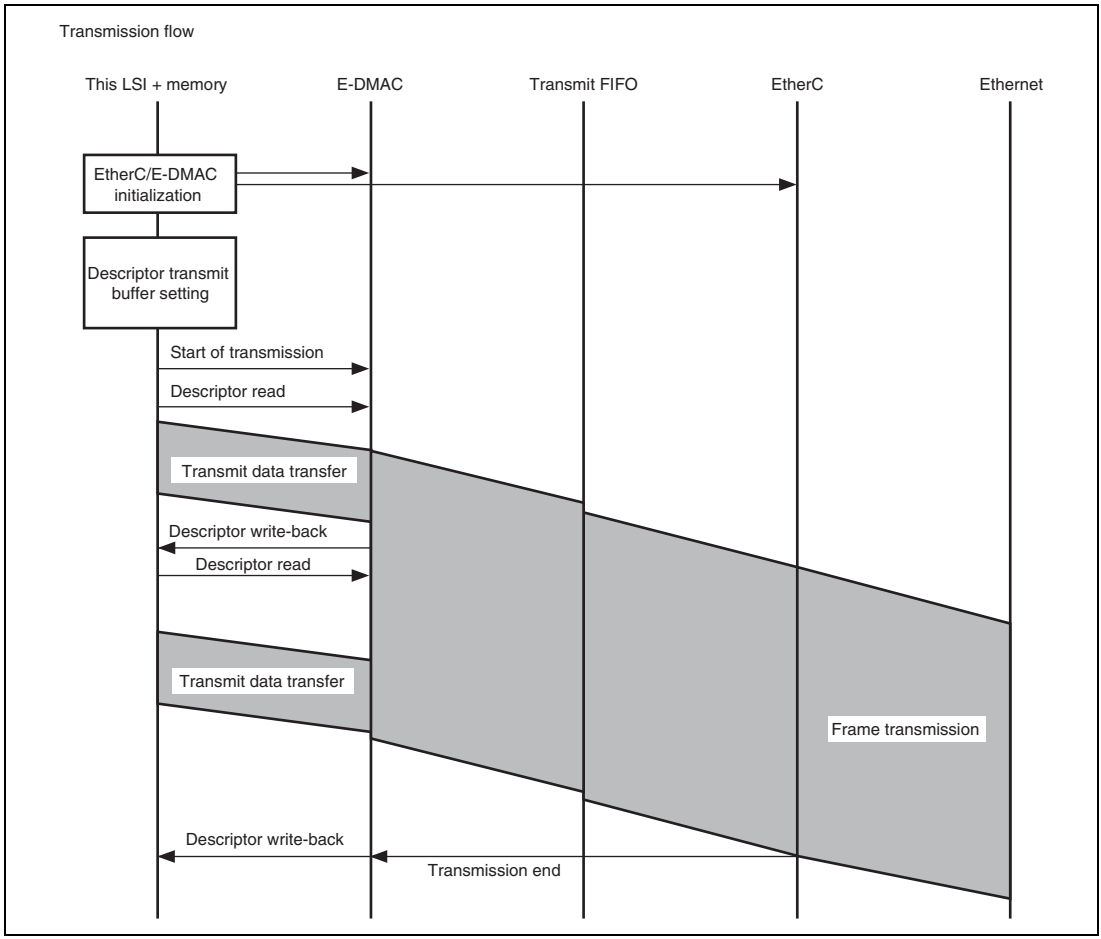


Figure 26.4 Example of Transmission Flow

26.3.3 Reception

When the CPU sets the receive request bit (RR) in the E-DMAC receive request register (EDRRR) while the receive function is enabled, the E-DMAC 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) in the initial state), and then enters the receiving standby state. Upon receiving a frame for own station while the RACT bit is set to 1 (valid), the E-DMAC 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 E-DMAC performs a write-back operation to the descriptor (with RFP set to 10 or 00) when the buffer becomes full, and then reads the next descriptor. The E-DMAC continues to transfer data to the receive buffer specified by the new RD2. When frame reception is completed, or if frame reception is suspended because of a certain kind of error, the E-DMAC performs write-back to the relevant descriptor (with RFP set to 11 or 01), and then ends the receive processing. The E-DMAC then reads the next descriptor and enters the receiving standby state again.

To receive frames continuously, the RNC bit in the receiving method control register (RMCR) must be set to 1. The initial value of the RNC bit is 0.

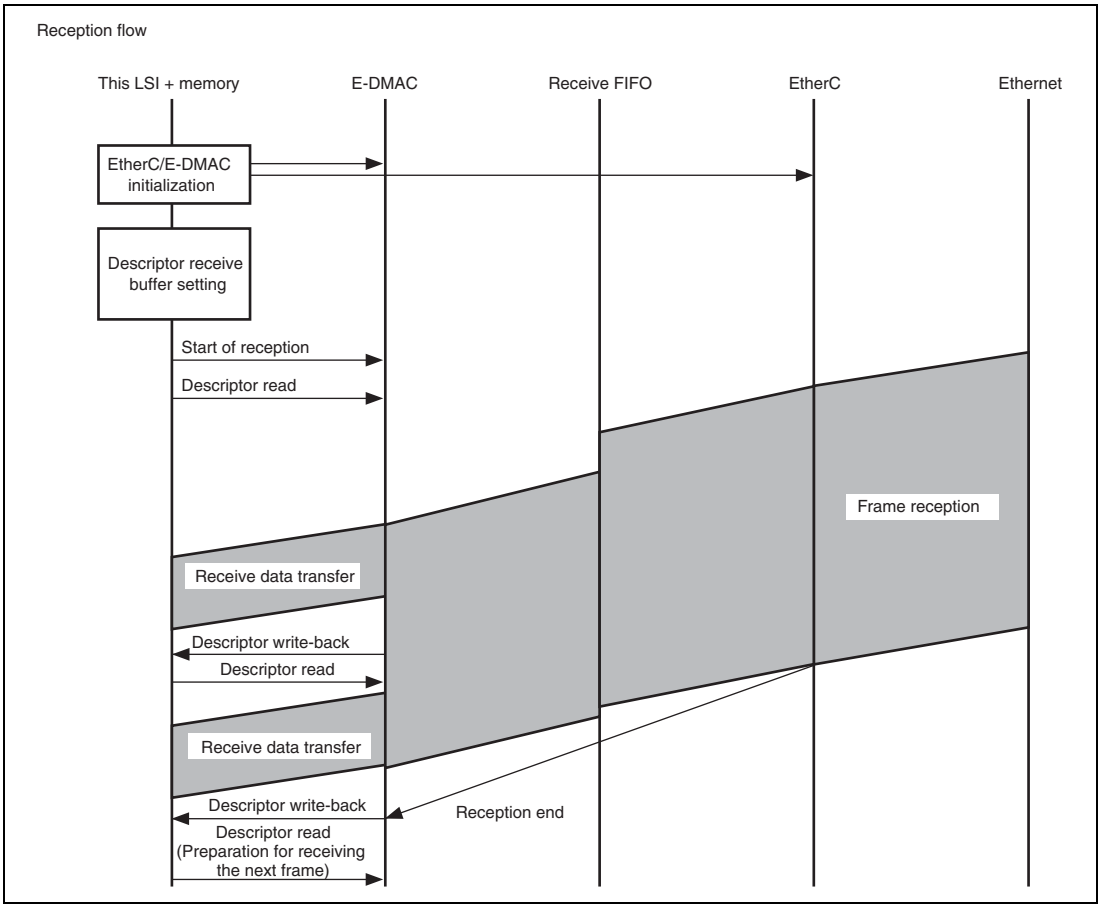


Figure 26.5 Example of Reception Flow

26.3.4 Transmit/Receive Processing of Multi-Buffer Frame

(1) Multi-Buffer Frame Transmit Processing

If an error occurs during multi-buffer frame transmission, the E-DMAC performs the processing shown in figure 26.6.

In the figure where the transmit descriptor is shown as inactive (TACT bit = 0), buffer data has already been transmitted normally, and where the transmit descriptor is shown as active (TACT bit = 1), buffer data has not been transmitted. If a frame transmit error occurs in the first descriptor part where the transmit descriptor is active (TACT bit = 1), transmission is halted and the TACT bit is cleared to 0 immediately. The next descriptor is then read, and the position within the transmit frame is determined on the basis of bits TFP1 and TFP0 (continuing [B'00] or end [B'01]). In the case of a continuing descriptor, the TACT bit is cleared to 0 only, and the next descriptor is read immediately. If the descriptor is the final descriptor, the TACT bit is cleared to 0 and write-back is also performed to the TFE and TFS bits at the same time. Data in the buffer is not transmitted during a period from the occurrence of an error until the write-back to the final descriptor. If error interrupts are enabled in the EtherC/E-DMAC status interrupt enable register (EESIPR), an interrupt is generated immediately after the final descriptor write-back.

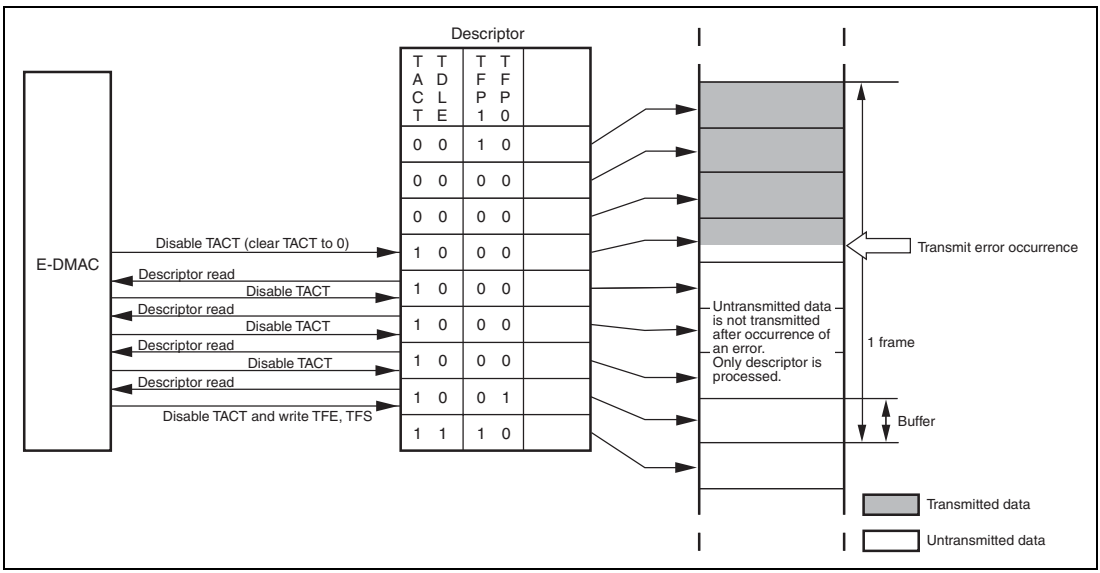


Figure 26.6 E-DMAC Operation after Transmit Error

(2) Multi-Buffer Frame Receive Processing

If an error occurs during reception of a multi-buffer frame, the E-DMAC performs the processing shown in figure 26.7.

In the figure, the invalid receive descriptors (RACT = 0) represent the normal reception of buffer data, and the valid receive descriptors (RACT = 1) represent unreceived buffers. If a frame receive error occurs in the first descriptor part where the RCAT bit is set to 1, the status is written back to the descriptor.

If error interrupts are enabled in the EtherC/E-DMAC status interrupt enable register (EESIPR), an interrupt is generated immediately after the write-back. If there is a new frame receive request, reception is continued from the buffer after that in which the error occurred.

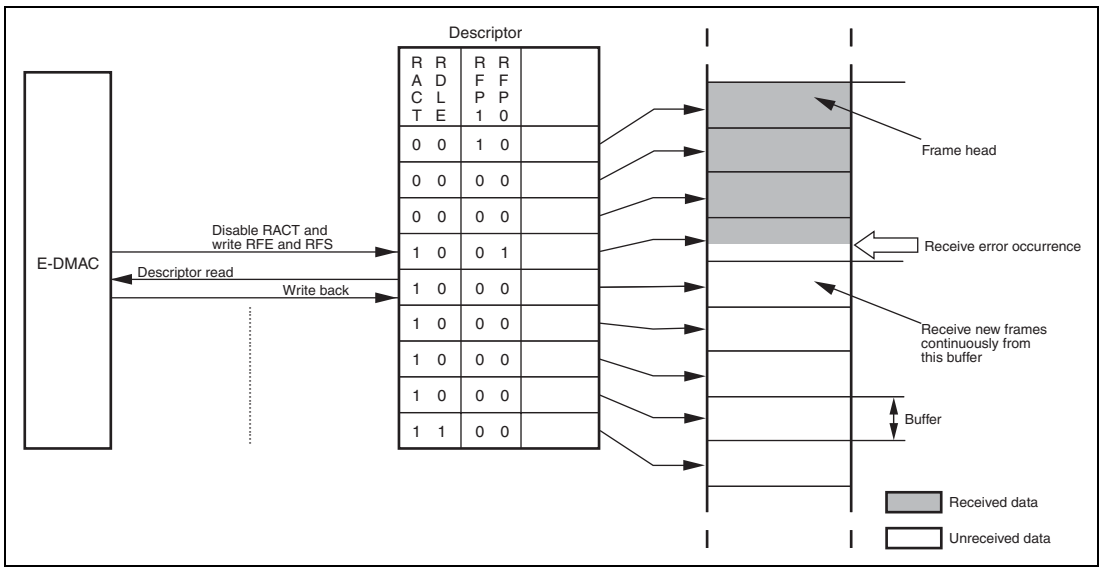


Figure 26.7 E-DMAC Operation after Receive Error

26.4 Usage Notes

(1) Number of Cycles for Access to Registers

Note that the number of cycles for access to E-DMAC registers differs from the number for access to registers in other on-chip peripheral modules (see section 9.5.12 (3), On-Chip Peripheral Module Access).

Section 27 Flash Memory (ROM)

The SH7214 and SH7216 Groups incorporate up to 1 Mbyte of flash memory (ROM) for the storage of instruction code. The ROM has the following features.

27.1 Features

- Two types of flash-memory MATs

  The ROM has two types of memory areas (hereafter referred to as memory MATs) in the same address space. These two MATs can be switched by the start-up mode or bank switching through the control register. For addresses H'00008000 to H'000FFFFF, undefined data is read and programming and erasing are ignored when the user boot MAT is selected.

  User MAT: 1 Mbyte (SH72167, SH72147)

  : 768 Kbytes (SH72166, SH72146)

  : 512 Kbytes (SH72165, SH72145)

  User boot MAT: 32 Kbytes

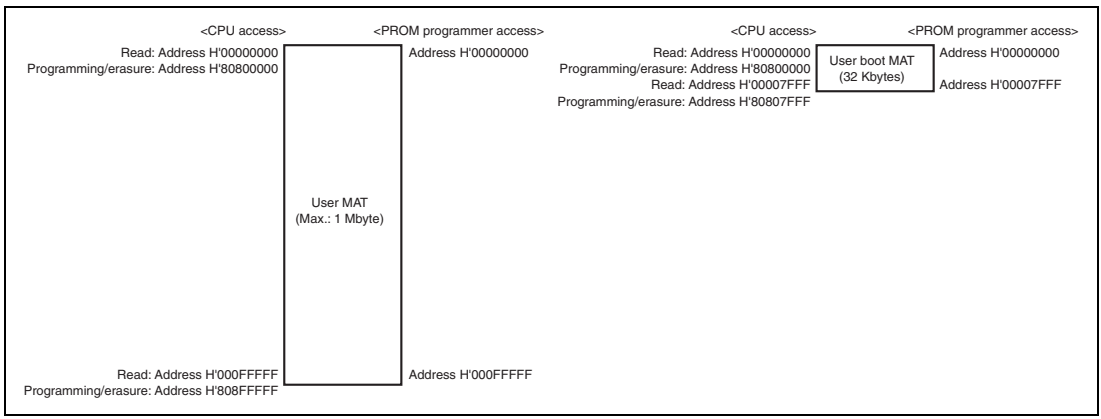


Figure 27.1 Memory MAT Configuration in ROM

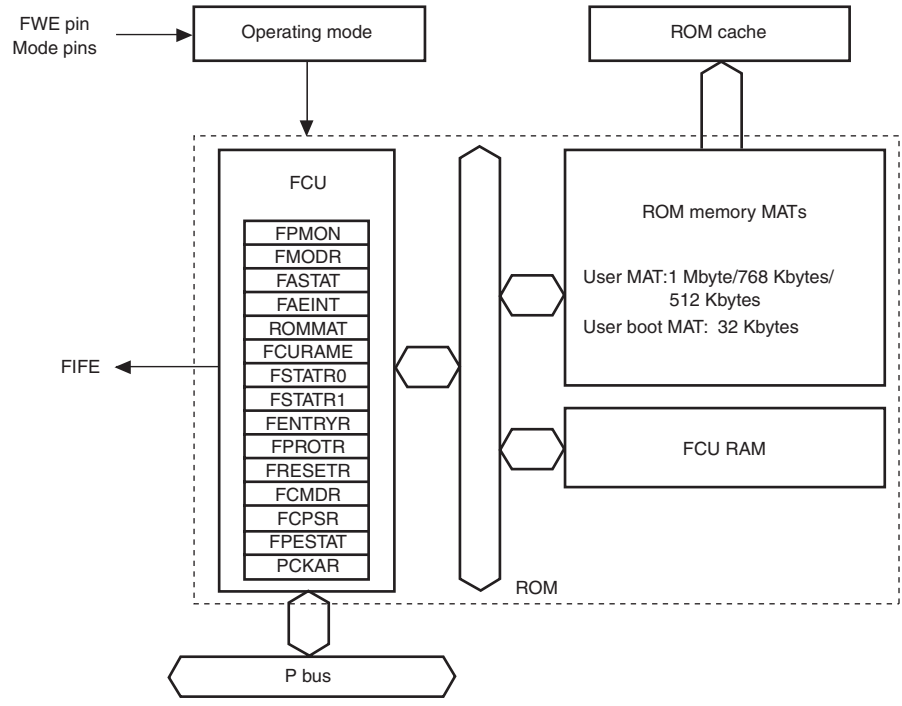
- High-speed reading through ROM cache

  Both the user MAT and user boot MAT can be read at high speed through the ROM cache. They can be read only in on-chip ROM enabled mode.

- Programming and erasing methods

  The ROM can be programmed and erased by commands issued through the peripheral bus (P bus) to the ROM/data flash (FLD) dedicated sequencer (FCU).

  While the flash control unit (FCU) is programming or erasing the ROM, the CPU can execute a program located outside the ROM. While the FCU is programming or erasing the FLD, the CPU can execute a program in the ROM. When the FCU suspends programming or erasure, the CPU can execute a program in the ROM, and then the FCU can resume programming or erasure. While the FCU suspends erasure, areas other than the erasure-suspended area can be programmed.



[Legend]

FPMON:	Flash pin monitor register
FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
ROMMAT:	ROM MAT select register
FCURAME:	FCU RAM enable register
FSTATR0, FSTATR1:	Flash status registers 0 and 1
FENTRYR:	Flash P/E mode entry register
FPROTR:	Flash protect register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FCPSR:	FCU processing switch register
FPESTAT:	Flash P/E status register
PCKAR:	Peripheral clock notification register
FIFE:	Flash interface error interrupt

Figure 27.2 Block Diagram of ROM

- Programming/erasing unit

  The user MAT and user boot MAT are programmed in 256-byte units. The entire area of the user boot MAT is always erased at one time. The user MAT can be erased in block units if the mode is not programmer mode. The entire area of the user MAT is erased in programmer mode.

  Figure 27.3 shows the block configuration of the user MAT. The user MAT is divided into eight 8-Kbyte blocks, nine 64-Kbyte blocks, and three 128-Kbyte blocks.

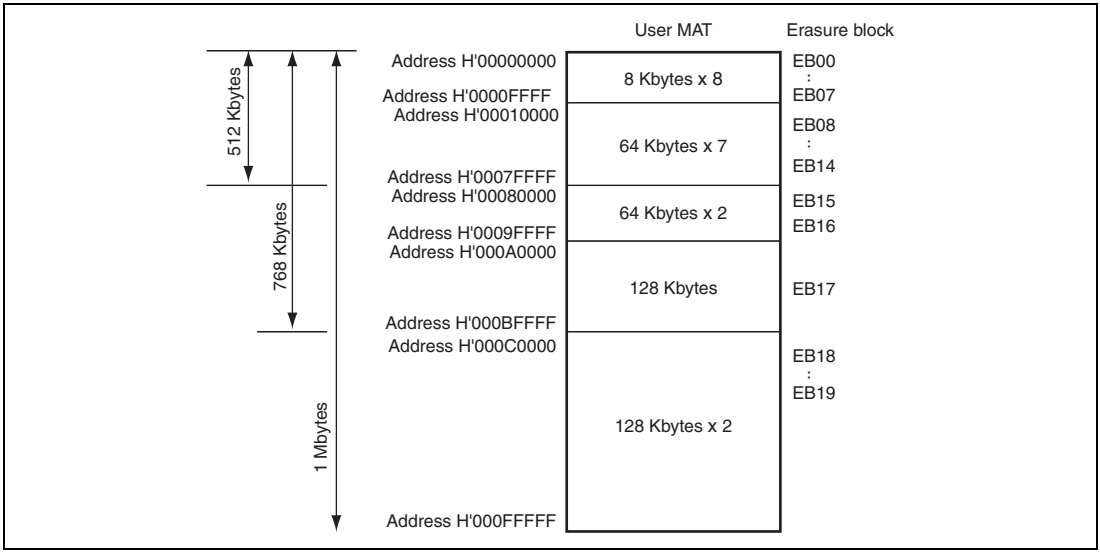


Figure 27.3 Block Configuration of User MAT

- Four types of on-board programming modes
  - — Boot mode

    The user MAT and user boot MAT can be programmed using the SCI. The bit rate for SCI communications between the host and this LSI can be automatically adjusted.
  - — USB boot mode

    The user MAT and user boot MAT can be programmed in program mode using the USB.
  - — User program mode

    The user MAT can be programmed with a desired interface. A transition from MCU mode 2 (MCU extended mode) or mode 3 (MCU single-chip mode) to this mode is enabled simply by changing the level on the FWE pin.
  - — User boot mode

    The user MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.
- One type of off-board programming mode
  - — Programmer mode

    The user MAT and user boot MAT can be programmed in programmer mode using the PROM programmer.
- Protection modes

  This LSI supports two modes to protect memory against programming or erasure: hardware protection by the levels on the FWE and mode pins and software protection by the FENTRY0 bit in FENTRYR or lock bit settings. The FENTRY0 bit enables or disables ROM programming or erasure by the FCU. A lock bit is included in each erasure block of the user MAT to protect memory against programming or erasure.

  The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure.
- Programming and erasing time and count

  Refer to section 33, Electrical Characteristics.

27.2 Input/Output Pins

Table 27.1 shows the input/output pins used for the ROM. The combination of MD1 and MD0 pin levels and the FWE pin level determines the ROM programming mode (see section 27.4, Overview of ROM-Related Modes). In boot mode, the ROM can be programmed or erased by the host connected via the PA3/RxD1 and PA4/TxD1 pins (see section 27.5, Boot Mode).

Table 27.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	This LSI enters the power-on reset state when this signal goes low.
Mode	MD1, MD0	Input	These pins specify the operating mode.
Flash programming enable	FWE	Input	This pin enables or disables ROM programming.
Receive data in SCI channel 1	PA3/RxD1	Input	Receives data through SCI channel 1 (communications with host)
Transmit data in SCI channel 1	PA4/TxD1	Output	Transmits data through SCI channel 1 (communications with host)
Pull-up control	PUPD (PB15)	Output	Pull-up control (used in USB boot mode)
USB data	USD+ USD-	I/O	USD signal from the USB that has a transceiver (used in USB boot mode)
USB cable connection monitor	VBUS	Input	Detects connection and disconnection of the USB cable (used in USB boot mode)
USB clock select	PB14	Input	Selects the clock supplied by the USB (used in USB boot mode)

27.3 Register Descriptions

Table 27.2 shows the ROM-related registers. Some of these registers have data flash (FLD) related bits, but this section only describes the ROM-related bits. For the FLD-related bits, refer to section 28.3, Register Descriptions. The ROM-related registers are initialized by a power-on reset.

Table 27.2 Register Configuration

Register Name	Symbol	R/W*[1]	Initial Value	Address	Access Size
Flash pin monitor register	FPMON	R	H'00 H'80	H'FFFFA800	8
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*[2]	H'00	H'FFFFA810	8
Flash access error interrupt enable register	FAEINT	R/W	H'9F	H'FFFFA811	8
ROM MAT select register	ROMMAT	R/(W)*[3]	H'0000 H'0001	H'FFFFA820	8, 16
FCU RAM enable register	FCURAME	R/(W)*[3]	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*[5]	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*[5]	H'FFFFA901	8, 16
Flash P/E mode entry register	FENTRYR	R/(W)*[4]	H'0000*[5]	H'FFFFA902	8, 16
Flash protect register	FPROTR	R/(W)*[4]	H'0000*[5]	H'FFFFA904	8, 16
Flash reset register	FRESETR	R/(W)*[3]	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*[5]	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W	H'0000*[5]	H'FFFFA918	8, 16
Flash P/E status register	FPESTAT	R	H'0000*[5]	H'FFFFA91C	8, 16
ROM cache control register	RCCR	R/W	H'00000001	H'FFFC1400	32
Peripheral clock notification register	PCKAR	R/W	H'0000*[5]	H'FFFFA938	8, 16

Notes: 1. In on-chip ROM disabled mode, the ROM-related registers are always read as 0 and writing to them is ignored.
2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.

4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.
5. These registers can be initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

27.3.1 Flash Pin Monitor Register (FPMON)

FPMON monitors the FWE pin state. FPMON is read as H'00 in on-chip ROM disabled mode. FPMON is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	FWE	—	—	—	—	—	—	—
Initial value:	1/0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FWE	1/0	R	Flash Write Enable Monitors the FWE pin level. The initial value depends on the FWE pin level when the LSI is started. 0: Disables ROM programming and erasure 1: Enables ROM programming and erasure
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.3.2 Flash Mode Register (FMODR)

FMODR specifies the FCU operation mode. In on-chip ROM disabled mode, FMODR is read as H'00 and writing to it is ignored. FMODR is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FRDMD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	FRDMD	0	R/W	FCU Read Mode Select Selects the read mode to read the ROM or FLD using FCU. This bit specifies the check method for the lock bits in the ROM (see section 27.6.1, FCU Command List, and section 27.6.3 (13), Reading Lock Bit), whereas this bit must be set to make the blank check command available for use in the FLD (see section 28, Data Flash (FLD)). 0: Selects the memory area read mode. The mode to read the lock bits in the ROM in ROM lock bit read mode. 1: Selects the register read mode. The mode to read the lock bits in the ROM using the lock bit read 2 command.
3 to 0	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.

27.3.3 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and FLD. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 27.9.3, Error Protection). To cancel a command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	Access Error Indicates whether or not a ROM access error has been generated. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters a command-locked state. 0: No ROM access error has occurred. 1: A ROM access error has occurred. [Setting conditions] • An access command is issued to ROM program/erase addresses H'80800000 to H'808FFFFF while the FENTRY0 bit in FENTRYR is 1 in ROM P/E normal mode. • An access command is issued to ROM program/erase addresses H'80800000 to H'808FFFFF while the FENTRY0 bit in FENTRYR is 0. • A read access command is issued to ROM read addresses H'00000000 to H'000FFFFF while the FENTRYR register value is not H'0000. • A block erase, program, or lock bit program command is issued while the user boot MAT is selected.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	• An access command is issued to an address other than ROM program/erase addresses H'80800000 to H'80807FFF while the user boot MAT is selected. [Clearing condition] • A 0 is written to this bit after reading a 1 from the ROMAE bit.
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLK	0	R	FCU Command Lock Indicates whether the FCU is in command-locked state (see section 27.9.3, Error Protection). 0: The FCU is not in a command-locked state 1: The FCU is in a command-locked state [Setting condition] • The FCU detects an error and enters command-locked state. [Clearing condition] • The FCU completes the status-clear command processing while FASTAT is H'10.
3	EEPAE	0	R/(W)*	FLD Access Error Refer to section 28, Data Flash (FLD).
2	EEPIFE	0	R/(W)*	FLD Instruction Fetch Error Refer to section 28, Data Flash (FLD).
1	EEPRPE	0	R/(W)*	FLD Read Protect Error Refer to section 28, Data Flash (FLD).
0	EEPWPE	0	R/(W)*	FLD Program/Erase Protect Error Refer to section 28, Data Flash (FLD).

Note: * Only 0 can be written to clear the flag after 1 is read.

27.3.4 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupts. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
Initial value:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAEIE	1	R/W	ROM Access Error Interrupt Enable Enables or disables an FIFE interrupt request when a ROM access error occurs and the ROMAE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when ROMAE = 1. 1: Generates an FIFE interrupt request when ROMAE = 1.
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLKIE	1	R/W	FCU Command Lock Interrupt Enable Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when CMDLK = 1 1: Generates an FIFE interrupt request when CMDLK = 1
3	EEPAEIE	1	R/W	FLD Access Error Interrupt Enable Refer to section 28, Data Flash (FLD).
2	EEPIFEIE	1	R/W	FLD Instruction Fetch Error Interrupt Enable Refer to section 28, Data Flash (FLD).
1	EEPRPEIE	1	R/W	FLD Read Protect Error Interrupt Enable Refer to section 28, Data Flash (FLD).

Bit	Bit Name	Initial Value	R/W	Description
0	EEPWPEIE	1	R/W	FLD Program/Erase Protect Error Interrupt Enable Refer to section 28, Data Flash (FLD).

27.3.5 ROM MAT Select Register (ROMMAT)

ROMMAT switches memory MATs in the ROM. In on-chip ROM disabled mode, ROMMAT is read as H'0000 and writing to it is ignored. ROMMAT is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	ROMSEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable ROMSEL bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ROMSEL	0/1	R/W	ROM MAT Select Selects a memory MAT in the ROM. The initial value is 1 when the LSI is started in user boot mode; otherwise, the initial value is 0. Writing to this bit is enabled only when this register is accessed in word size and H'3B is written to the KEY bits. 0: Selects the user MAT 1: Selects the user boot MAT

Note: * Write data is not retained.

27.3.6 FCU RAM Enable Register (FCURAME)

FCURAME enables or disables access to the FCU RAM area. In on-chip ROM disabled mode, FCURAME is read as H'00 and writing to it is ignored. FCURAME is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	—	—	—	FCRME
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	H'00	R/(W)*	Key Code These bits enable or disable FCRME bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FCRME	0	R/W	FCU RAM Enable Enables or disables access to the FCU RAM. Writing to this bit is enabled only when this register is accessed in word size and H'C4 is written to the KEY bits. Before writing to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. 0: Disables access to FCU RAM 1: Enables access to FCU RAM

Note: * Write data is not retained.

27.3.7 Flash Status Register 0 (FSTATR0)

FSTATR0 indicates the FCU status. In on-chip ROM disabled mode, FSTATR0 is read as H'00. FRTATR0 is initialized by a power-on reset, or setting the FRESET bit of the FRESETR register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	FRDY	ILG LERR	ERS ERR	PRG ERR	SUS RDY	—	ERS SPD	PRG SPD
Initial value:	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FRDY	1	R	Flash Ready Indicates the processing state in the FCU. 0: Programming or erasure processing, programming or erasure suspension processing, lock bit read 2 command processing, or FLD blank check is in progress (see section 28, Data Flash (FLD)). 1: None of the above is in progress.
6	ILGLERR	0	R	Illegal Command Error Indicates that the FCU has detected an illegal command or illegal ROM or FLD access. When this bit is 1, the FCU is in command-locked state (see section 27.9.3, Error Protection). 0: The FCU has not detected any illegal command or illegal ROM/FLD access 1: The FCU has detected an illegal command or illegal ROM/FLD access [Setting conditions] • The FCU has detected an illegal command. • The FCU has detected an illegal ROM/FLD access (the ROMAE, EEPAE, EEPIFE, EEPRPE, or EEPWPE bit in FASTAT is 1). • The FENTRYR setting is illegal. [Clearing condition] • The FCU completes the status-clear command processing while FASTAT is H'10.

Bit	Bit Name	Initial Value	R/W	Description
5	ERSERR	0	R	Erasure Error Indicates the result of ROM or FLD erasure by the FCU. When this bit is 1, the FCU is in command-locked state (see section 27.9.3, Error Protection). 0: Erasure processing has been completed successfully 1: An error has occurred during erasure [Setting conditions] • An error has occurred during erasure. • A block erase command has been issued for the area protected by a lock bit. [Clearing condition] • The FCU completes the status-clear command processing.
4	PRGERR	0	R	Programming Error Indicates the result of ROM or FLD programming by the FCU. When this bit is 1, the FCU is in command-locked state (see section 27.9.3, Error Protection). 0: Programming has been completed successfully 1: An error has occurred during programming [Setting conditions] • An error has occurred during programming. • A programming command has been issued for the area protected by a lock bit. [Clearing condition] • The FCU completes the status-clear command processing.

Bit	Bit Name	Initial Value	R/W	Description
3	SUSRDY	0	R	Suspend Ready Indicates whether the FCU is ready to accept a P/E suspend command. 0: The FCU cannot accept a P/E suspend command 1: The FCU can accept a P/E suspend command [Setting condition] • After initiating programming/erasure, the FCU has entered a state where it is ready to accept a P/E suspend command. [Clearing conditions] • The FCU has accepted a P/E suspend command. • The FCU has entered a command-locked state during programming or erasure.
2	—	0	R	Reserved This bit is always read as 0. Correct operation is not guaranteed if 1 is written to this bit.
1	ERSSPD	0	R	Erasure-Suspended Status Indicates that the FCU has entered an erasure suspension process or an erasure-suspended status (see section 27.6.4, Suspending Operation). 0: The FCU is in a status other than the below-mentioned. 1: The FCU is in an erasure suspension process or an erasure-suspended status. [Setting condition] • The FCU has initiated an erasure suspend command. [Clearing condition] • The FCU has accepted a resume command.

Bit	Bit Name	Initial Value	R/W	Description
0	PRGSPD	0	R	Programming-Suspended Status Indicates that the FCU has entered a write suspension process or a write suspend status (see section 27.6.4, Suspending Operation). 0: The FCU is in a status other than the below-mentioned. 1: The FCU is in a write suspension process or a write-suspended status. [Setting condition] • The FCU has initiated a write suspend command. [Clearing condition] • The FCU has accepted a resume command.

27.3.8 Flash Status Register 1 (FSTATR1)

FSTATR1 indicates the FCU status. In on-chip ROM disabled mode, FSTATR1 is read as H'00. FSTATR1 is initialized by a power-on reset, or setting the FRESET bit of the FRESETR register is set to 1.

Bit:	7	6	5	4	3	2	1	0
	FCUERR	—	—	FLOCKST	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	FCUERR	0	R	FCU Error Indicates an error has occurred during the CPU processing in the FCU. 0: No error has occurred during the CPU processing in the FCU 1: An error has occurred during the CPU processing in the FCU [Clearing condition] • The FRESET bit in FRESETR is set to 1. When FCUERR is 1, set the FRESET bit to 1 to initialize the FCU, and then copy the FCU firmware again from the FCU firmware area to the FCU RAM area.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	FLOCKST	0	R	Lock Bit Status Reflects the lock bit data read through lock bit read 2 command execution. When the FRDY bit becomes 1 after the lock bit read 2 command is issued, valid data is stored in this bit. This bit value is retained until the next lock bit read 2 command is completed. 0: Protected state 1: Non-protected state

Bit	Bit Name	Initial Value	R/W	Description
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

27.3.9 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or FLD. To specify the P/E mode for the ROM or FLD so that the FCU can accept commands, set either of FENTRYD and FENTRY0 bits to 1. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR can be initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

In access to the FENTRYR for a mode transition of the FCU, write to the register and then read it, and only proceed with programming, erasure or reading of the ROM after confirming the register setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FEKEY								FENTRYD	—	—	—	—	—	—	FENTRY0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FEKEY	All 0	R/(W)*	Key Code These bits enable or disable rewriting of the FENTRYD and FENTRY0 bits. Data written to these bits are not retained.
7	FENTRYD	0	R/W	FLD P/E Mode Entry Bit Refer to section 28, Data Flash (FLD).
6 to 1	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
0	FENTRY0	0	R/W	ROM P/E Mode Entry Bit 0 These bits specify the P/E mode for the 1-Mbyte ROM (read addresses: H'00000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF). 0: the 1-Mbyte ROM is in read mode 1: the 1-Mbyte ROM is in P/E mode Programming is enabled when the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FWE bit in FPMON is 1. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRY while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • The FRDY bit in FSTATR0 becomes 1 and the FWE bit in FPMON becomes 0. • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRY while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.

Note: * Write data is not retained.

27.3.10 Flash Protect Register (FPROTR)

FPROTR enables or disables the protection function through the lock bits against programming and erasure. In on-chip ROM disabled mode, FPROTR is read as H'0000 and writing to it is ignored. FPROTR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FPKEY								—	—	—	—	—	—	—	FPROTCN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FPKEY	H'00	R/(W)*	Key Code These bits enable or disable FPROTCN bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FPROTCN	0	R/W	Lock Bit Protect Cancel Enables or disables protection through the lock bits against programming and erasure. 0: Enables protection through the lock bits 1: Disables protection through the lock bits [Setting condition] • H'55 is written to FPKEY and 1 is written to FPROTCN in word access while the FENTRYR register value is not H'0000. [Clearing conditions] • This register is written to in byte access. • A value other than H'55 is written to FPKEY in word access. • H'55 is written to FPKEY and 0 is written to FPROTCN in word access. • The FENTRYR register value is H'0000.

Note: * Write data is not retained.

27.3.11 Flash Reset Register (FRESETR)

FRESETR is used for the initialization of FCU. In on-chip ROM disabled mode, FRESETR is read as H'0000 and writing to it is ignored. FRESETR is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FRKEY								—	—	—	—	—	—	—	FRESET
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FRKEY	H'00	R/(W)*	Key Code These bits enable or disable FRESET bit modification. The data written to these bits are not stored.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	FRESET	0	R/W	Flash Reset Setting this bit to 1 forcibly terminates programming/erasure of ROM or FLD and initializes the FCU. A high voltage is applied to the ROM/FLD memory units during programming and erasure. To ensure sufficient time for the voltage applied to the memory unit to drop, keep the value of the FRESET bit at 1 for a period of t_{RESW2} (see section 33, Electrical Characteristics) when the FCU is initialized. Do not read from the ROM/FLD units while the value of the FRESET bit is kept at 1. The FCU commands are unavailable for use while the FRESET bit is set to 1, since this initializes the FENTRYR register. This bit can be written only when H'CC is written to FRKEY in word access. 0: Issue no reset to the FCU. 1: Issues a reset to the FCU.

Note: * Write data is not retained.

27.3.12 FCU Command Register (FCMDR)

FCMDR stores the commands that the FCU has accepted. In on-chip ROM disabled mode, FCMDR is read as H'0000 and writing to it is ignored. FCMDR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMDR								PCMDR							
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	CMDR	H'FF	R	Command Register These bits store the latest command accepted by the FCU.
7 to 0	PCMDR	H'FF	R	Precommand Register These bits store the previous command accepted by the FCU.

Table 27.3 shows the states of FCMDR after acceptance of the various commands. For details on the blank check, see section 28.6, User Mode, User Program Mode, and User Boot Mode.

Table 27.3 FCMDR Status after a Command is Accepted

Command	CMDR	PCMDR
Normal mode transition	H'FF	Previous command
Status read mode transition	H'70	Previous command
Lock bit read mode transition (lock bit read 1)	H'71	Previous command
Program	H'E8	Previous command
Block erase	H'D0	H'20
P/E suspend	H'B0	Previous command
P/E resume	H'D0	Previous command
Status register clear	H'50	Previous command
Lock bit read 2 blank check	H'D0	H'71
Lock bit program	H'D0	H'77
Peripheral clock notification	H'E9	Previous command

27.3.13 FCU Processing Switch Register (FCPSR)

FCPSR selects a function to make the FCU suspend erasure. In on-chip ROM enabled mode, FCPSR is read as H'0000 and writing to it is ignored. FCPSR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSPMD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ESUSPMD	0	R/W	Erasure-Suspended Mode Selects the erasure-suspended mode to be entered when a P/E suspend command is issued while the FCU is erasing the ROM or FLD (see section 27.6.4, Suspending Operation). 0: Suspension-priority mode 1: Erasure-priority mode

27.3.14 Flash P/E Status Register (FPESTAT)

FPESTAT indicates the result of programming/erasure of the ROM/FLD. In on-chip ROM enabled mode, FPESTAT is read as H'0000 and writing to it is ignored. FPESTAT is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PEERRST							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	PEERRST	H'00	R	P/E Error Status Indicates the source of an error that occurs during programming/erasure. This bit value is only valid if the PRGERR or ERSERR bit value in FSTATR0 is 1; otherwise the bit retains the value to indicate the source of an error that previously occurred. H'01: A write attempt made to an area protected by the lock bits H'02: A write error caused by other source than the above H'11: An erase attempt made to an area protected by the lock bits H'12: An erase error caused by other source than the above Other than above: Reserved

27.3.15 ROM Cache Control Register (RCCR)

RCCR contains the RCF bit that controls the disabling of all lines in the ROM cache.

This register can be accessed only in longwords.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	RCF	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	RCF	0	R/W	ROM Cache Flush Writing a 1 to this bit disables (flushes) the instructions or data in the ROM cache. This bit is read as 0. 0: Does not disable the instructions or data in the ROM cache. 1: Disables the instructions or data in the ROM cache. [Clearing condition] • Reset/standby [Setting condition] • Writing a 1.
2, 1	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
0	—	1	R	Reserved The write value should always be 1; otherwise normal operation cannot be guaranteed.

27.3.16 Peripheral Clock Notification Register (PCKAR)

PCKAR is used to notify the sequencer of information regarding the frequency setting of the peripheral clock (Pϕ) for programming or erasure of the ROM or data flash memory. The setting governs the time programming or erasure takes. In modes where the internal ROM is disabled, the value read from the PCKAR will be H'0000 and writing to the PCKAR will be ineffective. PCKAR is initialized by a power-on reset or by writing 1 to the FRESET bit in FRESETR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCKA							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. When writing to the register, always write 0 to these bits. Operation is not guaranteed if 1 is written to any or all of these bits.
7 to 0	PCKA	H'00	R/W	Peripheral Clock Notification These bits are used to notify the peripheral clock (Pϕ) for programming or erasure of the ROM or data flash memory. Set the frequency of Pϕ by setting these bits before programming or erasure, and then issue a peripheral clock notification command. Do not change the frequency while the ROM or data flash memory is being programmed or erased. Follow the procedure below to calculate the setting. • Convert the frequency expressed in MHz units to binary notation, and write the value to the PCKA bits. For example, if the frequency of the peripheral clock is 35.9 MHz, the setting is derived as follows. • Round 35.9 up to obtain 36. • Convert 36 into binary form and set the PCKA bits to H'24 (B'00100100). Notes: 1. Do not issue the command for overwriting the ROM or data flash memory if the setting of the PCKA bits is for a frequency outside the range from 20 to 50 MHz. 2. If the frequency set by the PCKA bits differs from the actual frequency, there is a possibility of destroying the ROM or data flash memory.

27.4 Overview of ROM-Related Modes

Figure 27.4 shows the ROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD1, MD0 and FWE pin settings, refer to section 3, MCU Operating Modes.

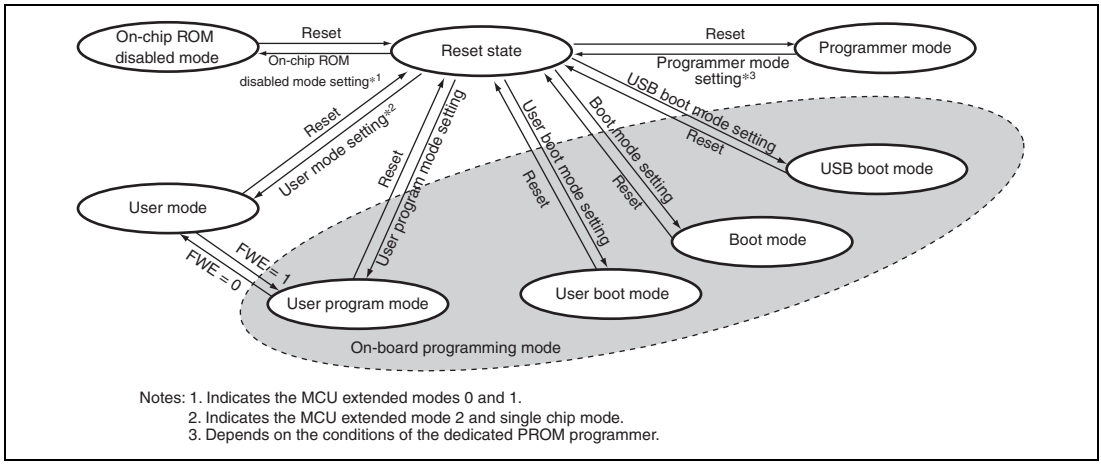


Figure 27.4 ROM-Related Mode Transition

- The ROM cannot be read, programmed, or erased in on-chip ROM disabled mode (MCU extended modes 0 and 1).
- The ROM can be read but cannot be programmed or erased in user mode (MCU extended mode 2 and single chip mode).
- The ROM can be read, programmed, and erased on the board in user program mode, user boot mode, boot mode, and USB boot mode.

Table 27.4 compares programming- and erasure-related items for the boot mode, user program mode, user boot mode, USB boot mode, and programmer mode.

Table 27.4 Comparison of Programming Modes

Item	Boot Mode	User Program Mode	User Boot Mode	USB Boot Mode	Programmer Mode
Programming/ erasure environment		On-board programming			Off-board programming
Programming/ erasure enabled MAT	User MAT and user boot MAT	User MAT	User MAT	User MAT and user boot MAT	User MAT and user boot MAT
Programming/ erasure control	Host	FCU	FCU	Host	Programmer
Entire area erasure	Available (automatic)	Available	Available	Available (automatic)	Available (automatic)
Block erasure	Available*[1]	Available	Available	Available*[1]	Not available
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM	From host via USB	Via programmer
Reset-start MAT	Embedded program stored MAT	User MAT	User boot MAT*[2]	Embedded program stored MAT	Embedded program stored MAT
Transition to MCU operating mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	Mode setting change and reset	—
Pin state	CK: output Other pins: input RxD1 (PA3) and TxD1 (PA4): valid (The same as the states in MCU extension mode 2)	Dependent on user settings	CK: output (initial setting) Other pins: input (initial setting)	CK: output Other pins: input (MCU extension mode 2)	Programmer dedicated pins

Notes: 1. The entire area is erased when the LSI is started. After that, a specified block can be erased.

2. After the LSI is started in the embedded program stored MAT and the boot program provided by Renesas Corp. is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode, USB boot mode, and programmer mode.
- In boot mode or USB boot mode, the user MAT, user boot MAT, and FLD data MAT are all erased immediately after the LSI is started. The user MAT, user boot MAT, and data MAT can then be programmed from the host via the SCI. The ROM can also be read after this entire area erasure.
- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user program mode.

27.5 Boot Mode

27.5.1 System Configuration

To program or erase the user MAT and user boot MAT in boot mode, send control commands and programming data from the host. The on-chip SCI of this LSI is used in asynchronous mode for communications between the host and this LSI. The tool for sending control commands and programming data must be prepared in the host. When this LSI is started in boot mode, the program in the embedded program stored MAT is executed. This program automatically adjusts the SCI bit rate and performs communications between the host and this LSI by means of the control command method.

Figure 27.5 shows the system configuration in boot mode. The NMI and $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ interrupts are ignored in this mode, but these pins must be fixed to non-active state. Note that the AUD cannot be used in this mode.

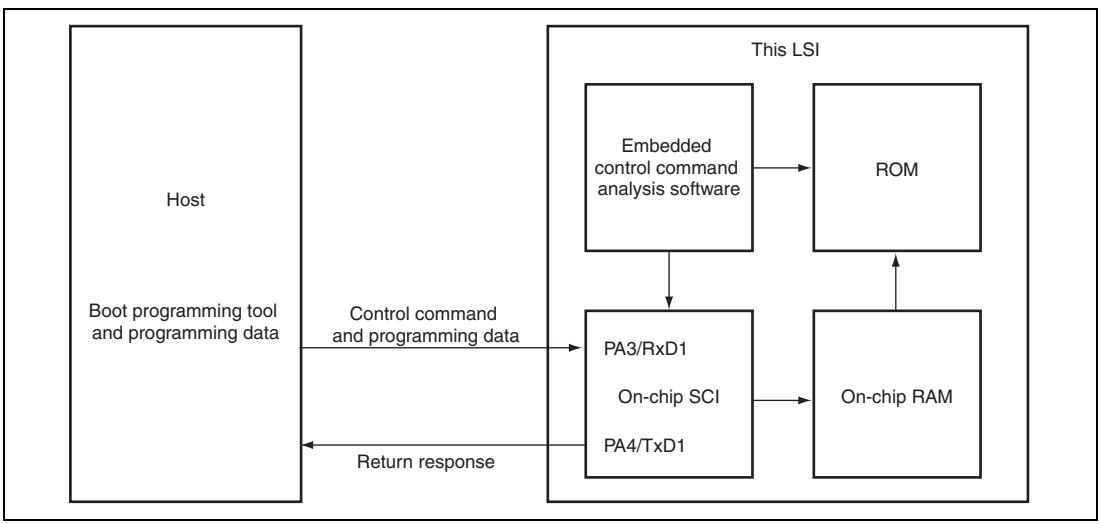


Figure 27.5 System Configuration in Boot Mode

27.5.2 State Transition in Boot Mode

Figure 27.6 shows the state transition in boot mode.

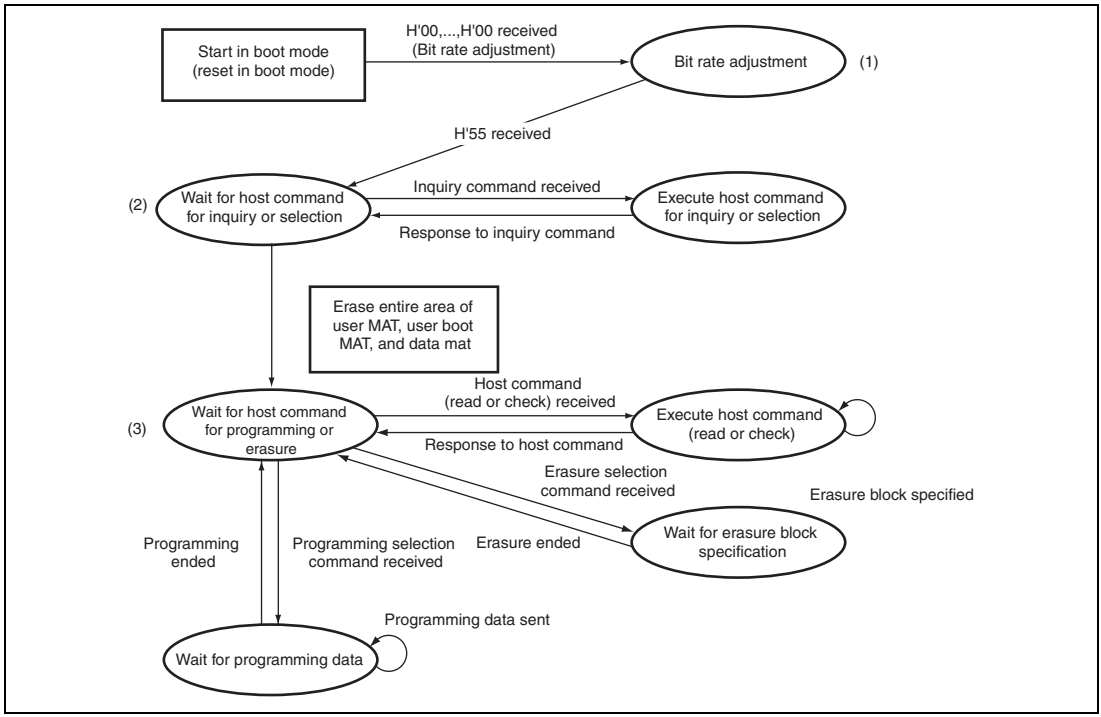


Figure 27.6 State Transition in Boot Mode

(1) Bit Rate Adjustment

After this LSI is started in boot mode, it automatically adjusts the bit rate for communications between the host and SCI. After automatic adjustment of the bit rate, the LSI sends H'00 to the host. After the LSI has successfully received H'55 sent from the host, the LSI waits for a host command for inquiry or selection. For details on bit rate adjustment, see section 27.5.3, Automatic Adjustment of Bit Rate.

(2) Waiting for Host Command for Inquiry or Selection

In this state, the host inquires regarding MAT information (such as the size, configuration, and start address) and the supported functions, and selects the device, clock mode, and bit rate. Upon reception of a programming/erasure state transition command sent from the host, this LSI erases the entire area of each of the user MAT, user boot MAT, and FLD data MAT and waits for a host command for programming or erasure. For details of inquiry/selection host commands, see section 27.5.5, Inquiry/Selection Host Command Wait State.

(3) Waiting for Host Command for Programming or Erasure

In this state, this LSI performs programming or erasure according to the command sent from the host. The LSI enters programming data wait state, erasure block specification wait state, or command (read or check) processing state depending on the received command.

Upon reception of a programming selection command, the LSI waits for programming data. After the programming selection command, send the programming start address and programming data from the host. Specifying H'FFFFFFFF as the programming start address terminates programming processing and the LSI makes a transition from the programming data wait state to programming/erasure command wait state.

Upon reception of an erasure selection command, the LSI waits for erasure block specification. After the erasure selection command, send the erasure block number from the host. Specifying H'FF as the erasure block number terminates erasure processing and the LSI makes a transition from the erasure block specification wait state to programming/erasure command wait state. As the entire area of each of the user MAT, user boot MAT, and FLD data MAT is erased before the LSI enters programming/erasure command wait state after it is started in boot mode, erasure processing is not needed except for the case when the data programmed in boot mode should be erased without resetting the LSI.

In addition to programming and erasing commands, many other host commands are provided for use in programming/erasure command wait state; these include commands for checksum, blank check (erasure check), memory read, and status inquiry. For details on these host commands, see section 27.5.6, Programming/Erasing Host Command Wait State.

27.5.3 Automatic Adjustment of Bit Rate

When this LSI is started in boot mode, it measures the low-level (H'00) period of the data that is continuously sent from the host in asynchronous SCI communications. During this measurement, set the SCI transmit/receive format to 8-bit data, 1 stop bit, and no parity, and set the bit rate to 9,600 bps or 19,200 bps. This LSI calculates the bit rate of the host SCI by means of the measured low-level period, and then sends H'00 to the host after completing the bit rate adjustment. When the host has received H'00 successfully, it must send H'55 to this LSI. If the host has failed to receive H'00, restart this LSI in boot mode to calculate and adjust the bit rate again. When this LSI has received H'55, it returns H'E6 to the host, or when it has failed to receive H'55, it returns H'FF.

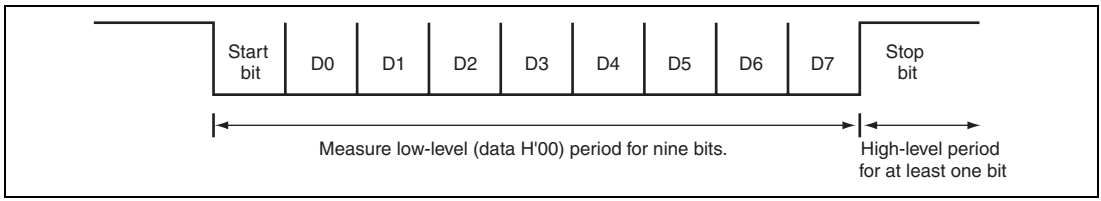


Figure 27.7 SCI Transmit/Receive Format for Automatic Adjustment of Bit Rate

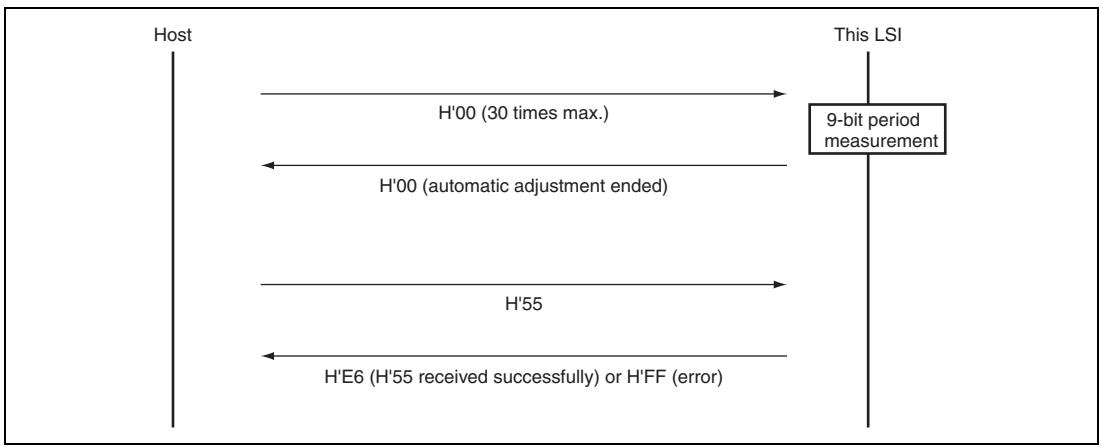


Figure 27.8 Communication Sequence between Host and This LSI

The bit rate may not be adjusted correctly depending on the bit rate of the host SCI or the peripheral clock frequency of this LSI. Satisfy the SCI communications condition as shown in table 27.5.

Table 27.5 Condition for Automatic Adjustment of Bit Rate

Host SCI Bit Rate	Peripheral Clock Frequency of this LSI
9,600 bps	20 to 50 MHz
19,200 bps	20 to 50 MHz

27.5.4 USB Boot Mode

USB boot mode is used to send control commands and programming data from the externally connected host via the USB to program and erase the user MAT and user boot MAT.

USB boot mode needs programming data on the host side and a tool to send control commands and programming data. Figure 27.9 shows the system configuration in USB boot mode. All interrupt requests generated in USB boot mode are ignored. No interrupt requests should be generated on the system side.

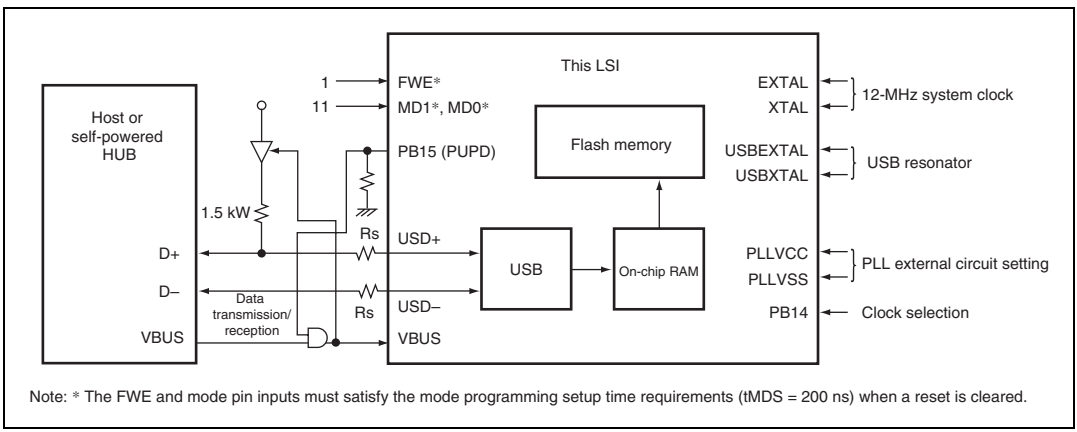


Figure 27.9 System Configuration in USB Boot Mode

(1) Features

- Power mode: Self-powered mode
- The D+ pull-up control connection is supported only by the PUPD pin (PB15).
- For enumeration information, see table 27.6.

Table 27.6 Enumeration Information

USB standard	Ver. 2.0 (full-speed)
Transfer mode	Transfer mode control (in, out), bulk (in, out)
Endpoint structure	EP0 Control (in, out) 16 bytes Configuration 1 └ InterfaceNumber 0 └ AlternateSetting 0 ├ EP1 Bulk (out) 64 bytes ├ EP2 Bulk (in) 64 bytes ├ EP3 Interrupt (in) 16 bytes ├ EP4 Bulk (out) 64 bytes ├ EP5 Bulk (in) 64 bytes ├ EP6 Interrupt (in) 16 bytes ├ EP7 Bulk (out) 64 bytes ├ EP8 Bulk (in) 64 bytes └ EP9 Interrupt (in) 16 bytes

(2) State Transition

Figure 27.10 shows the state transition after this LSI is started in USB boot mode.

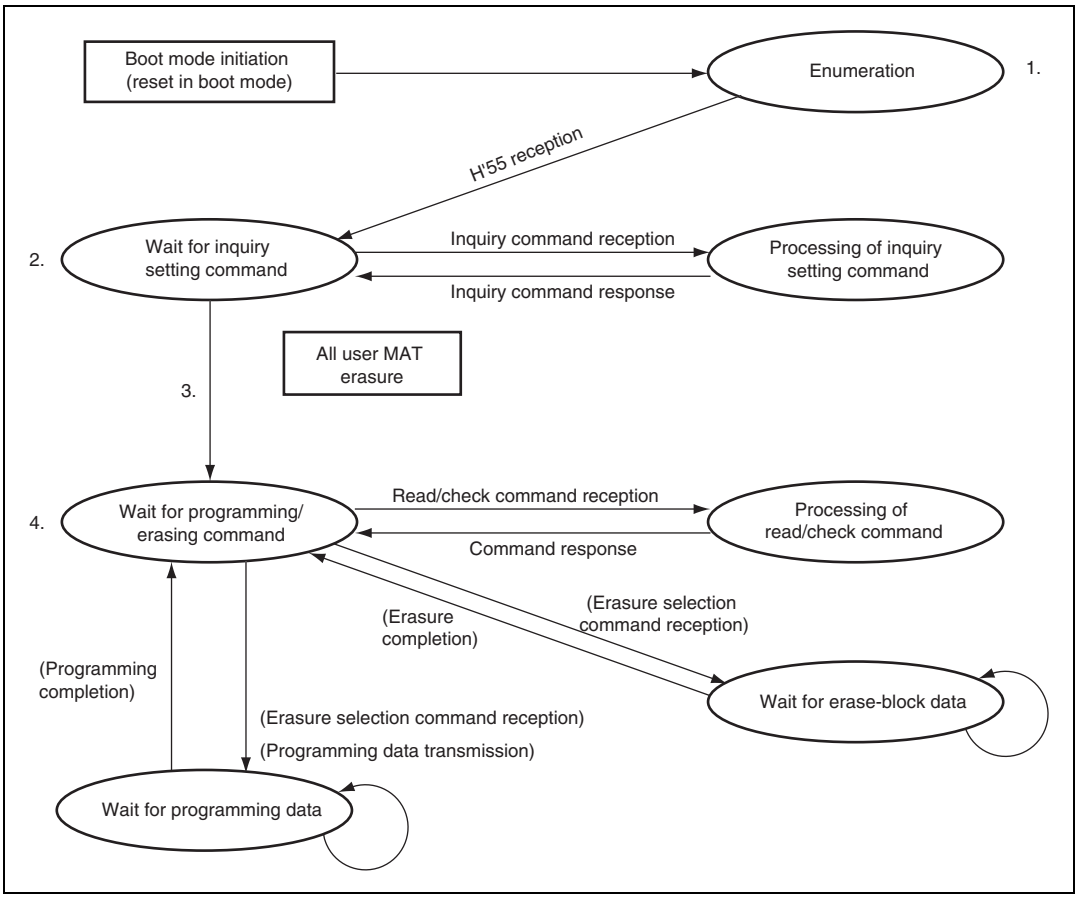


Figure 27.10 USB Boot Mode

1. When a transition to USB boot mode is made, the boot program embedded in this LSI is initiated. When the USB boot program is initiated, this LSI performs enumeration with the host. When enumeration is completed, the host transmits 1 byte of H'55 to this LSI. When the LSI cannot receive the byte normally, USB boot mode should be initiated again.
2. An inquiry about the size, configuration, start address, and support status of the user MAT and user boot MAT is transmitted to the host.
3. After inquiries have finished, all user MAT/user boot MAT are automatically erased.

4. After the user MAT/user boot MAT is erased automatically, the LSI waits for a programming/erasing command. After receiving a programming command, the LSI waits for programming data. The same applies to erasure.

   In addition to the programming/erasing command, there are the sum check command and blank check (erasure check) command of the user MAT/user boot MAT, and memory read command, and current status information acquisition command.

(3) Notes when Executing USB Boot Mode

- The USB module needs a 48-MHz clock. Set the frequency of the external clock and the clock oscillator to implement a 48-MHz USB-dedicated clock (Uϕ). For details, see section 4, Clock Pulse Generator (CPG).
- The PB14 pin is used to select the clock supplied to the USB.

  PB14 = 0: USBEXTAL or USBXTAL is used.

  PB14 = 1: The system clock is used.
- When PB14 = 0, connect a 48-MHz oscillator to USBEXTAL or USBXTAL.
- When PB14 = 1, connect a 12-MHz oscillator to EXTAL or XTAL with USBEXTAL = 0 and USBXTAL = open.
- For the D+ pull-up control connection, use the PUPD pin (PB15).
- To maintain stable power supply when programming or erasing flash memory, the cable should not be connected via the bus-powered hub.
- Note especially that unplugging the USB cable while the flash memory is being programmed or erased may destroy the LSI permanently in the worst case.

27.5.5 Inquiry/Selection Host Command Wait State

Table 27.7 shows the host commands available in inquiry/selection host command wait state. The boot program status inquiry command can also be used in programming/erasure host command wait state. The other commands can only be used in inquiry/selection host command wait state.

Table 27.7 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	Selects a 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
User MAT information inquiry	Inquires regarding the number of user 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
Simultaneous two-MAT programming information inquiry	Inquires regarding the availability of simultaneous two-MAT programming function
New bit rate selection	Modifies the bit rate of SCI communications between the host and this LSI
Programming/erasure state transition	Erases the entire area of each of the user MAT, user boot MAT, and FLD data MAT and makes this LSI enter programming/erasure host command wait state
Boot program status inquiry	Inquires regarding the state of this LSI

If the host has sent an undefined command, this LSI 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	H'80	Command

In 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 this LSI 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, this LSI returns a response indicating a command error. Figure 27.11 shows an example of the procedure to use inquiry/selection host commands.

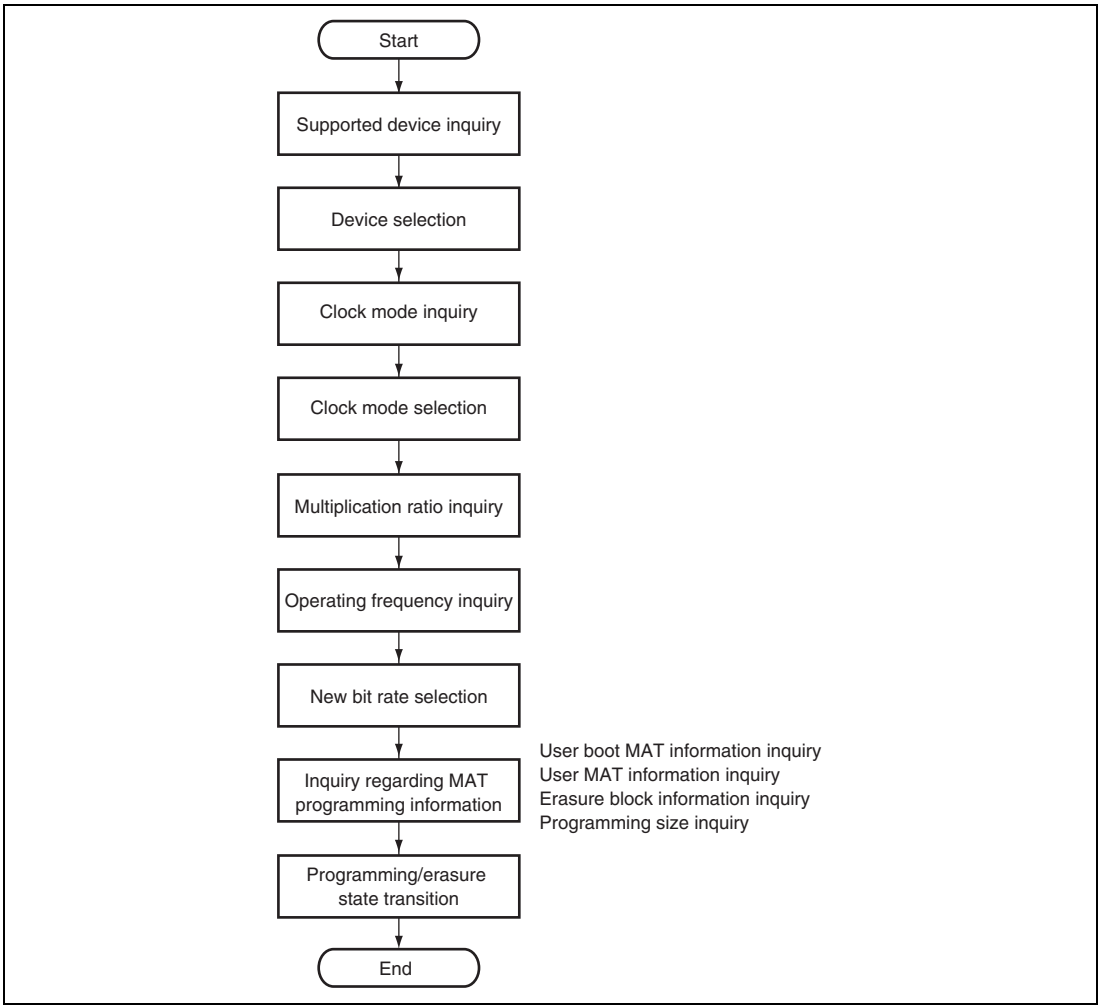


Figure 27.11 Example of Procedure to Use Inquiry/Selection Host Commands

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, this LSI 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, this LSI only returns the information concerning the selected device.

Command

H'20

Response

H'30	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

(2) Device Selection

In response to a device selection command sent from the command, this LSI checks if the selected device is supported. When the selected device is supported, this LSI specifies this device as the device for use and returns a response (H'06). If the selected device is not supported or the sent command is illegal, this LSI returns an error response (H'90).

Even when H'01 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	H'10	Size	Device code	SUM
Response	H'06			
Error response	H'90	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

H'11: Checksum error (illegal command)

H'21: Incorrect device code error

(3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, this LSI returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, this LSI only returns the information concerning the selected clock mode.

Command	H'21

Response				
	H'31	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, H'01 indicates clock mode 1)

SUM (1 byte): Checksum

(4) Clock Mode Selection

In response to a clock mode selection command sent from the host, this LSI checks if the selected clock mode is supported. When the selected mode is supported, this LSI specifies this clock mode for use and returns a response (H'06). If the selected mode is not supported or the sent command is illegal, this LSI returns an error response (H'91).

Be sure to issue a clock mode selection command only after issuing a device selection command. Even when H'00 or H'01 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	H'11	Size	Mode	SUM
Response	H'06			
Error response	H'91	Error		

[Legend]

Size (1 byte): Number of characters in the mode field (fixed at 1)

Mode (1 byte): Clock mode (one of the clock modes returned in response to the clock mode inquiry command)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error (illegal command)

H'22: Incorrect clock mode error

(5) Multiplication Ratio Inquiry

In response to a multiplication ratio inquiry command sent from the host, this LSI returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command

H'22

Response

H'32	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 count, and multiplication ratio fields

Clock type count (1 byte): Number of clock types (for example, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for example, H'03 indicates that three multiplication ratios are supported for the internal clock (x4, x6, and x8))

Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example, H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example, H'FE = -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, this LSI returns the minimum and maximum frequencies for each clock.

Command

H'23

Response

H'33	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, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Minimum frequency (2 bytes): Minimum value of the operating frequency (for example, H'07D0 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, this LSI returns the number of user boot MATs and their addresses.

Command

H'24

Response

H'34	Size	MAT count
MAT start address		
MAT end address		
MAT start address		
MAT end address		
:		
MAT start address		
MAT end address		
SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of user boot MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a user boot MAT

MAT end address (4 bytes): End address of a user boot MAT

SUM (1 byte): Checksum

(8) User MAT Information Inquiry

In response to a user MAT information inquiry command sent from the host, this LSI returns the number of user MATs and their addresses.

Command	H'25		
Response	H'35	Size	MAT count
	MAT start address		
	MAT end address		
	MAT start address		
	MAT end address		
	:		
	MAT start address		
	MAT end address		
	SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of user MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a user MAT

MAT end address (4 bytes): End address of a user MAT

SUM (1 byte): Checksum

(9) Erasure Block Information Inquiry

In response to an erasure block information inquiry command sent from the host, this LSI returns the number of erasure blocks in the user MAT and their addresses.

Command	H'26		
Response	H'36	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, this LSI returns the programming size.

Command	H'27			
Response	H'37	Size	Programming size	SUM

[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, this LSI 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, this LSI returns a response (H'06) 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, this LSI returns an error response (H'BF). Upon reception of response H'06, 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 bit rate to the new one. After that, the host sends confirmation data (H'06) in the new bit rate, and this LSI returns a response (H'06) to the confirmation data.

Be sure to issue a new bit rate selection command only after a clock mode selection command.

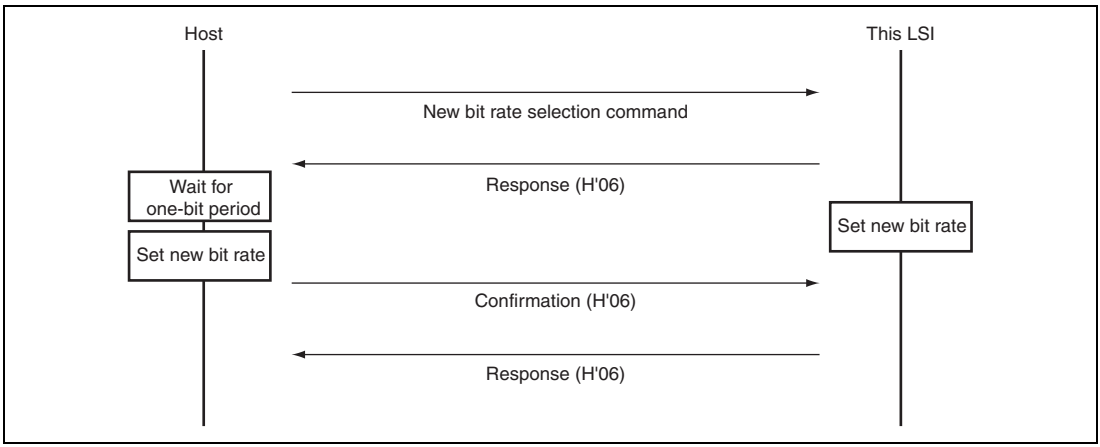


Figure 27.12 New Bit Rate Selection Sequence

Command	H'3F	Size	Bit rate	Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2	
	SUM			
Response	H'06			
Error response	H'BF	Error		
Confirmation	H'06			
Response	H'06			

[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, H'00C0 indicates 19200 bps)

1/100 of the new bit rate value should be specified.

Input frequency (2 bytes): Clock frequency input to this LSI (for example, H'07D0 indicates 20.00 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, H'02 indicates two clock types; that is, an internal clock and a peripheral clock)

Multiplication ratio 1 (1 byte): Multiplication/division ratio of the input frequency to obtain the internal clock

A positive value indicates a multiplication ratio (for example, H'04 = 4 = multiplication by 4)

A negative value indicates a division ratio (for example, HFE = -2 = division by 2)

Multiplication 2 (1 byte): Multiplication/division ratio of the input frequency to obtain the peripheral clock

This value is represented in the same format as multiplication ratio 1

SUM (1 byte): Checksum

Error: Error code

H'11: Checksum error

H'24: Bit rate selection error

H'25: Input frequency error

H'26: Multiplication ratio error

H'27: 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 this LSI 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 (fEX), multiplication ratio 2 (Pϕ), the SCBRR setting (N) in SCI, and the CKS[1:0] bit value (N) in SCSMR.

$$\text{Error (\%)} = \frac{f_{EX} \times P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1$$

- 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 this LSI cannot operate at the operating frequencies selected through a new bit rate selection command. This LSI 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/Erasure State Transition

In response to a programming/erasure state transition command sent from the host, this LSI erases the entire area of each of the user MAT, user boot MAT, and FLD data MAT. After completing erasure, this LSI returns a response (H'06) and waits for a programming/erasure host command. If this LSI has failed to complete erasure due to an error, it returns an error response (sends H'C0 and H'51 in that order).

Do not issue a programming/erasure state transition command before device selection, clock mode selection, and new bit rate selection commands.

Command	H'40	
Response	H'06	
Error response	H'C0	H'51

(13) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, this LSI returns its current status. The boot program status inquiry command can be issued in both inquiry/selection host command wait state and programming/erasure host command wait state.

Command	H'4F			
Response	H'5F	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 this LSI (see table 27.8)

Error (1 byte): Error status in this LSI (see table 27.9)

Table 27.8 Status Code

Code	Description
H'11	Waiting for device selection
H'12	Waiting for clock mode selection
H'13	Waiting for bit rate selection
H'1F	Waiting for transition to programming/erasure host command wait state (bit rate has been selected)
H'31	Erasing the user MAT and user boot MAT
H'3F	Waiting for a programming/erasure host command
H'4F	Waiting for reception of programming data
H'5F	Waiting for erasure block selection

Table 27.9 Error Code

Code	Description
H'00	No error
H'11	Checksum error
H'21	Incorrect device code error
H'22	Incorrect clock mode error
H'24	Bit rate selection error
H'25	Input frequency error
H'26	Multiplication ratio error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data size error
H'51	Erasure error
H'52	Incomplete erasure error
H'53	Programming error
H'54	Selection error
H'80	Command error
H'FF	Bit rate adjustment verification error

27.5.6 Programming/Erasing Host Command Wait State

Table 27.10 shows the host commands available in programming/erasure host command wait state.

Table 27.10 Programming/Erasure Host Commands

Host Command Name	Function
User boot MAT programming selection	Selects the program for user boot MAT programming
User MAT programming selection	Selects the program for user MAT programming
Simultaneous two-user MAT programming selection	Selects the program for simultaneous two-user MAT programming
256-byte programming	Programs 256 bytes of data
Erasure selection	Selects the erasure program
Block erasure	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 enabled	Enables the lock bit protect
Lock bit disable	Disables the lock bit protect
Boot program status inquiry	Inquires regarding the state of this LSI

If the host has sent an undefined command, this LSI returns a response indicating a command error. For the format of this response, see section 27.5.5, Inquiry/Selection Host Command Wait State.

To program the ROM, issue a programming selection command (user boot MAT programming selection or user MAT programming selection command) and then a 256-byte programming command from the host. Upon reception of a programming selection command, this LSI enters programming data wait state (see section 27.5.2, State Transition in Boot Mode). In response to a 256-byte programming command sent from the host in this state, this LSI starts programming the ROM. When the host sends a 256-byte programming command specifying H'FFFFFFFF as the

programming start address, this LSI detects it as the end of programming and enters programming/erasure host command wait state.

To erase the ROM, issue an erasure selection command and then a block erasure command from the host. Upon reception of an erasure selection command, this LSI enters erasure block selection wait state (see section 27.5.2, State Transition in Boot Mode). In response to a block erasure command sent from the host in this state, this LSI erases the specified block in the ROM. When the host sends a block erasure command specifying H'FF as the block number, this LSI detects it as the end of erasure and enters programming/erasure host command wait state.

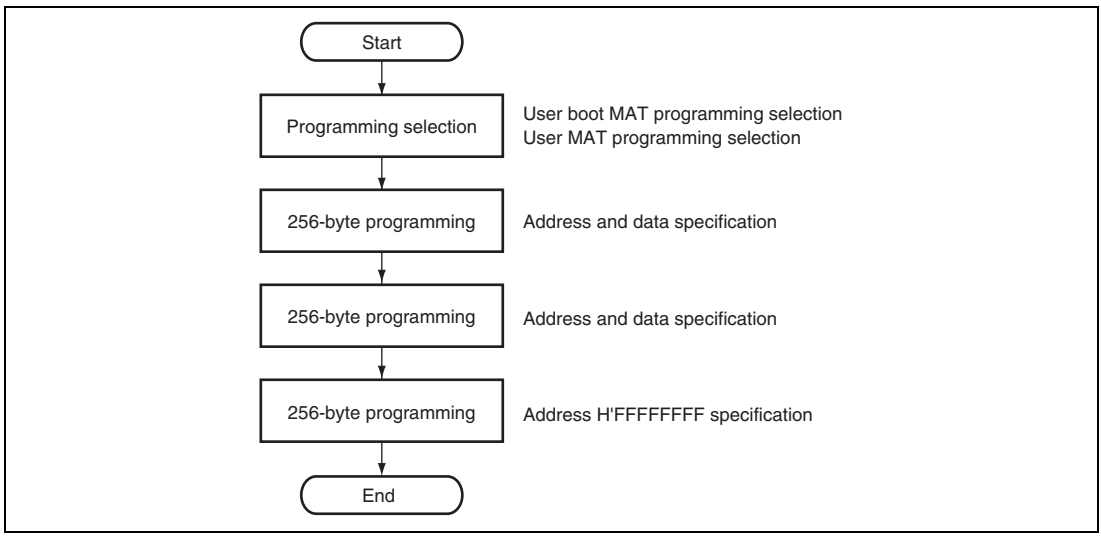


Figure 27.13 Procedure for ROM Programming in Boot Mode

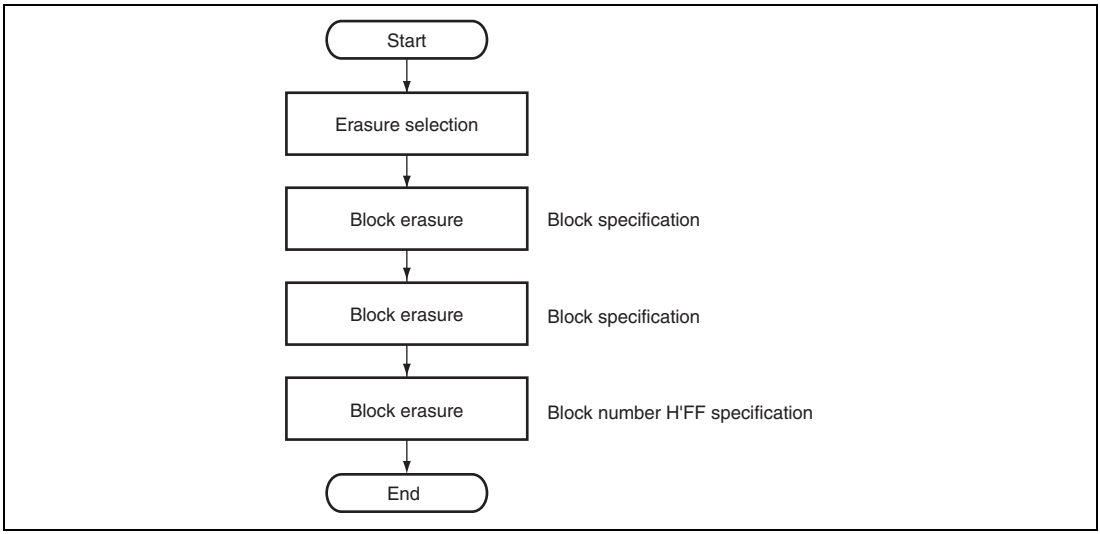


Figure 27.14 Procedure for ROM Erasure in Boot Mode

Each host command is described in detail below. The "command" in the description indicates a command sent from the host to this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) User Boot MAT Programming Selection

In response to a user boot MAT programming selection command sent from the host, this LSI selects the program for user boot MAT programming and waits for programming data.

Command	H'42
Response	H'06

(2) User MAT Programming Selection

In response to a user MAT programming selection command sent from the host, this LSI selects the program for user MAT programming and waits for programming data.

Command	H'43
Response	H'06

(3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, this LSI programs the ROM. After completing ROM programming successfully, this LSI returns a response (H'06). If an error has occurred during ROM programming, this LSI returns an error response (H'D0).

Command	H'50	Programming Address		
	Data	Data	. . .	Data
	SUM			
Response	H'06			
Error response	H'D0	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, H'FFFFFFFF should be specified.

Data (256 bytes): Programming data
H'FF should be specified for the bytes that do not need to be programmed.
When terminating programming, no data needs to be specified (only the programming address and SUM should be sent in that order).

SUM (1 byte): Checksum

Error (1 byte): Error code
H'11: Checksum error
H'2A: Address error (the specified address is not in the target MAT)
H'53: Programming cannot be done due to a programming error

(4) Erasure Selection

In response to an erasure selection command sent from the host, this LSI selects the erasure program and waits for erasure block specification.

Command	H'48
Response	H'06

(5) Block Erasure

In response to a block erasure command sent from the host, this LSI erases the ROM. After completing ROM erasure successfully, this LSI returns a response (H'06). If an error has occurred during ROM erasure, this LSI returns an error response (H'D8).

Command	H'58	Size	Block	SUM
Response	H'06			
Error response	H'D8	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, H'FF should be specified.

SUM (1 byte): Checksum

Error (1 byte): Error code
H'11: Checksum error
H'29: Block number error (an incorrect block number is specified)
H'51: Erasure cannot be done due to an erasure error

(6) Memory Read

In response to a memory read command sent from the host, this LSI reads data from the ROM. After completing ROM reading, this LSI returns the data stored in the address specified by the memory read command. If this LSI has failed to read the ROM, this LSI returns an error response (H'D2).

Command	H'52	Size	Area	Read start address	
	Reading size				SUM

Response	H'52	Reading size		
	Data	Data	. . .	Data
	SUM			

Error response	H'D2	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
H'00: User boot MAT
H'01: 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
H'11: Checksum error
H'2A: Address error
- The value specified for area selection is neither H'00 nor H'01.
- The specified read start address is outside the selected MAT.

H'2B: Data size error
- H'00 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, this LSI sums the user boot MAT data in byte units and returns the result (checksum).

Command	H'4A			
Response	H'5A	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, this LSI sums the user MAT data in byte units and returns the result (checksum).

Command	H'4B			
Response	H'5B	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, this LSI checks whether the user boot MAT is completely erased. When the user boot MAT is completely erased, this LSI returns a response (H'06). If the user boot MAT has an unerased area, this LSI returns an error response (sends H'CC and H'52 in that order).

Command	H'4C	
Response	H'06	
Error response	H'CC	H'52

(10) User MAT Blank Check

In response to a user MAT blank check command sent from the host, this LSI checks whether the user MAT is completely erased. When the user MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'CD and H'52 in that order).

Command	H'4D	
Response	H'06	
Error response	H'CD	H'52

(11) Read Lock Bit Status

In response to a read lock bit status command sent from the host, this LSI reads data from the lock bit. After completing the lock bit reading, this LSI returns the data stored in the address specified by the read lock bit status command. If this LSI has failed to read the lock bit, this LSI returns an error response (H'F1).

Command	H'71	Size	Area	Medium address	Upper address	SUM
Response	Status					
Error response	H'F1	Error				

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3 in this LSI)

Area (1 byte): Target MAT to be read

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Status (1 byte): Bit 6 locked at "0"

Bit 6 unlocked at "1"

Error (1 byte): Error code

H'11: Checksum error

H'2A: 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, this LSI writes to a lock bit and locks the specified block. After completing the lock bit blocking, this LSI returns a response (H'06). If this LSI has failed to lock, this LSI returns an error response (H'F7).

Command	H'77	Size	Area	Medium address	Upper address	SUM
Response	H'06					
Error response	H'F7	Error				

[Legend]

Size (1 byte): Total number of bytes in the area, medium address, and upper address (fixed at 3 in this LSI)

Area (1 byte): Target MAT to be locked

H'00: User boot MAT

H'01: User MAT

Medium address (1 byte): Medium address at the end of the specified address (8 to 15 bits)

Upper address (1 byte): Upper address at the end of the specified address (16 to 23 bits)

SUM (1 byte): Checksum

Error (1 byte): Error code

H'11: Checksum error

H'2A: Address error (the specified address is not in the target MAT)

H'53: 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, this LSI enables a lock bit.

Command	H'7A
Response	H'06

(14) Lock Bit Disable

In response to a lock bit enable command sent from the host, this LSI disables a lock bit.

Command	H'75
Response	H'06

(15) Boot Program Status Inquiry

For details, refer to section 27.5.5, Inquiry/Selection Host Command Wait State.

27.6 User Program Mode

27.6.1 FCU Command List

To program or erase the user MAT in user program mode, issue FCU commands to the FCU. Table 27.11 is a list of FCU commands for ROM programming and erasure.

Table 27.11 FCU Command List (ROM-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 27.6.2, Conditions for FCU Command Acceptance)
Status read mode transition	Moves to the status read mode (see section 27.6.2, Conditions for FCU Command Acceptance)
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 27.6.2, Conditions for FCU Command Acceptance)
Program	Programs ROM (in 256-byte units)
Block erase	Erases ROM (in block units; erasing the lock bit)
P/E suspend	Suspends programming or erasure
P/E resume	Resumes programming or erasure
Status register clear	Clears the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state
Lock bit read 2	Reads the lock bit of a specified erasure block (updates the FLOCKST bit in FSTATR1 to reflect the lock bit state)
Lock bit program	Writes to the lock bit of a specified erasure block
Peripheral clock notification	Specifies the peripheral clock frequency

FCU commands other than the lock bit read 2 program and lock bit program are also used for FLD programming and erasure. When a lock bit read 2 command is issued to the FLD, an FLD blank check is executed. When a lock bit program command is issued to the FLD, it is detected as an illegal command and generates an error (see section 28, Data Flash (FLD)).

To issue a command to the FCU, write to a ROM program/erase address through the P bus. Table 27.12 shows the FCU command format. Performing P-bus write access as shown in table 27.12 under specified conditions starts each command processing in the FCU. For the conditions for FCU command acceptance, refer to section 27.6.2, Conditions for FCU Command Acceptance. For details of each FCU command, refer to section 27.6.3, FCU Command Usage.

When H'71 is sent in the first cycle of an FCU command while the FRDMD bit is 0 (memory area read mode), the FCU accepts the lock bit read mode transition command (lock bit read 1). When a ROM program/erase address is read through the P bus after transition to the lock bit read mode, the FCU copies the lock bit of the erasure block corresponding to the accessed address into all bits in the read data. When H'71 is sent in the first cycle of the FCU command while the FRDMD bit is 1 (register read mode), the FCU waits for the second-cycle data (H'D0) of the lock bit read 2 command. When a ROM program/erase address is written to through the P bus in this state, the FCU copies the lock bit of the erasure block corresponding to the accessed address into the FLOCKST bit in FSTATR1.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 27.6.4, Suspending Operation.

Table 27.12 FCU Command Format

Command	Number of Bus Cycles	First Cycle		Second Cycle		Third Cycle		Fourth and Fifth Cycles		Sixth Cycle		Seventh to 130th Cycles		131st Cycle	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Normal mode transition	1	RA	H'FF	—	—	—	—	—	—	—	—	—	—	—	—
Status read mode transition	1	RA	H'70	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	RA	H'71	—	—	—	—	—	—	—	—	—	—	—	—
Program	131	RA	H'E8	RA	H'80	WA	WD1	RA	WDn	RA	WDn	RA	WDn	RA	H'D0
Block erase	2	RA	H'20	BA	H'D0	—	—	—	—	—	—	—	—	—	—
P/E suspend	1	RA	H'B0	—	—	—	—	—	—	—	—	—	—	—	—
P/E resume	1	RA	H'D0	—	—	—	—	—	—	—	—	—	—	—	—
Status register clear	1	RA	H'50	—	—	—	—	—	—	—	—	—	—	—	—
Lock bit read 2	2	RA	H'71	BA	H'D0	—	—	—	—	—	—	—	—	—	—
Lock bit program	2	RA	H'77	BA	H'D0	—	—	—	—	—	—	—	—	—	—
Peripheral clock notification	6	RA	H'E9	RA	H'03	WA	H'0F0F	WA	H'0F0F	RA	H'D0	—	—	—	—

[Legend]

RA: ROM program/erase address
An address in the range from H'80800000 to H'808FFFFF

WA: ROM program address
Start address of 256-byte programming data

BA: ROM erasure block address
An address in the target erasure block (specified by the ROM program/erase address)

WDn: n-th word of programming data (n = 1 to 128)

27.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 27.15 is an FCU mode transition diagram.

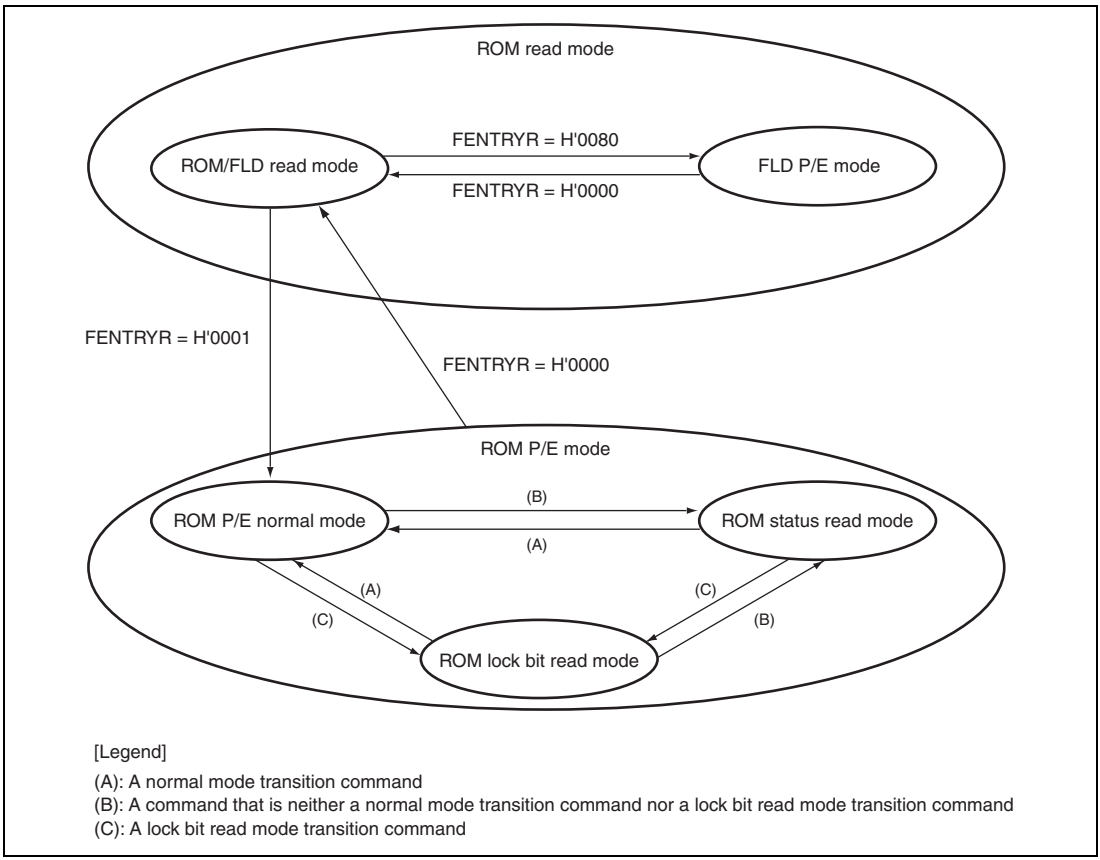


Figure 27.15 FCU Mode Transition Diagram (ROM-Related Modes)

(1) ROM Read Mode

- ROM/FLD read mode

  The ROM and FLD can be read through the ROM cache and HPB, respectively, at a high speed. 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 to 0 in FENTRYR.

- FLD P/E mode

  The ROM can be read through the ROM cache at a high speed. The FCU accepts commands for FLD, but does not accept commands for ROM. The FCU enters this mode when the FENTRY0 bit is set to 0 and the FENTRYD bit to 1. For details of the FLD P/E mode, refer to section 27.6.2, Conditions for FCU Command Acceptance.

(2) ROM P/E Mode

- ROM P/E normal mode

  The FCU enters this mode when the FENTRYD bit is set to 0 and the FENTRY0 bit is set to 1 in ROM read mode, or when a normal mode transition command is accepted in ROM P/E mode. Table 27.13 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, a ROM access error occurs and the FCU enters the command-locked state (see section 27.9.3, Error Protection).

- ROM status read mode

  The FCU enters this mode when the FCU accepts a command that is neither a normal mode transition command nor a lock bit read mode transition command in ROM P/E mode. The ROM status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 27.13 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. If an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, the FSTATR0 value is read.

- ROM lock bit read mode

  The FCU enters this mode when the FCU accepts a lock bit read mode transition command in ROM P/E mode. Table 27.13 shows the commands that can be accepted in this mode. High-speed read operation is not available for the ROM. The FENTRYR value is the same as that in ROM P/E normal mode. If an address in the range from H'80800000 to H'808FFFFF is read through the P-bus while the FENTRY0 bit is set to 1, the lock bit value of the target erasure block is returned through all bits in the read data.

Table 27.13 shows the acceptable commands in each FCU mode/state. When a command that cannot be accepted is issued, the FCU enters the command-locked state (see section 27.9.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR bit value in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR bit value in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 27.13, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and lock bit read 2 processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 27.13 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 27.13 FCU Modes/States and Acceptable Commands

	P/E Normal Mode			Status Read Mode							Lock Bit Read Mode		
Item	Programming-Suspended	Erasure-Suspended	Other State	Programming/Erasure Processing	Programming/Erasure Suspension Processing	Lock Bit Read 2 Processing	Programming-Suspended	Erasure-Suspended	Command-Locked	Other State	Programming-Suspended	Erasure-Suspended	Other State
FRDY bit in FSTATR0	1	1	1	0	0	0	1	1	0/1	1	1	1	1
SUSRDY bit in FSTATR0	0	0	0	1	0	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0	0	0/1	0	0	1	0	0	0	1	0
PRGSPD bit in FSTATR0	1	0	0	0	0/1	0	1	0	0	0	1	0	0
CMDLK bit in FASTAT	0	0	0	0	0	0	0	0	1	0	0	0	0
Normal mode transition	A	A	A	×	×	×	A	A	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	A	A	×	A	A	A	A
Lock bit read mode transition (lock bit read 1)	A	A	A	×	×	×	A	A	×	A	A	A	A
Program	×	*	A	×	×	×	×	*	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	A	A	×	×	A	A	×
Status register clear	A	A	A	×	×	×	A	A	A	A	A	A	A
Lock bit read 2	A	A	A	×	×	×	A	A	×	A	A	A	A
Lock bit program	×	*	A	×	×	×	×	*	×	A	×	*	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	A	×	×	A

[Legend]

A: Acceptable

*: Only programming is acceptable for the areas other than the erasure-suspended block

×: Not acceptable

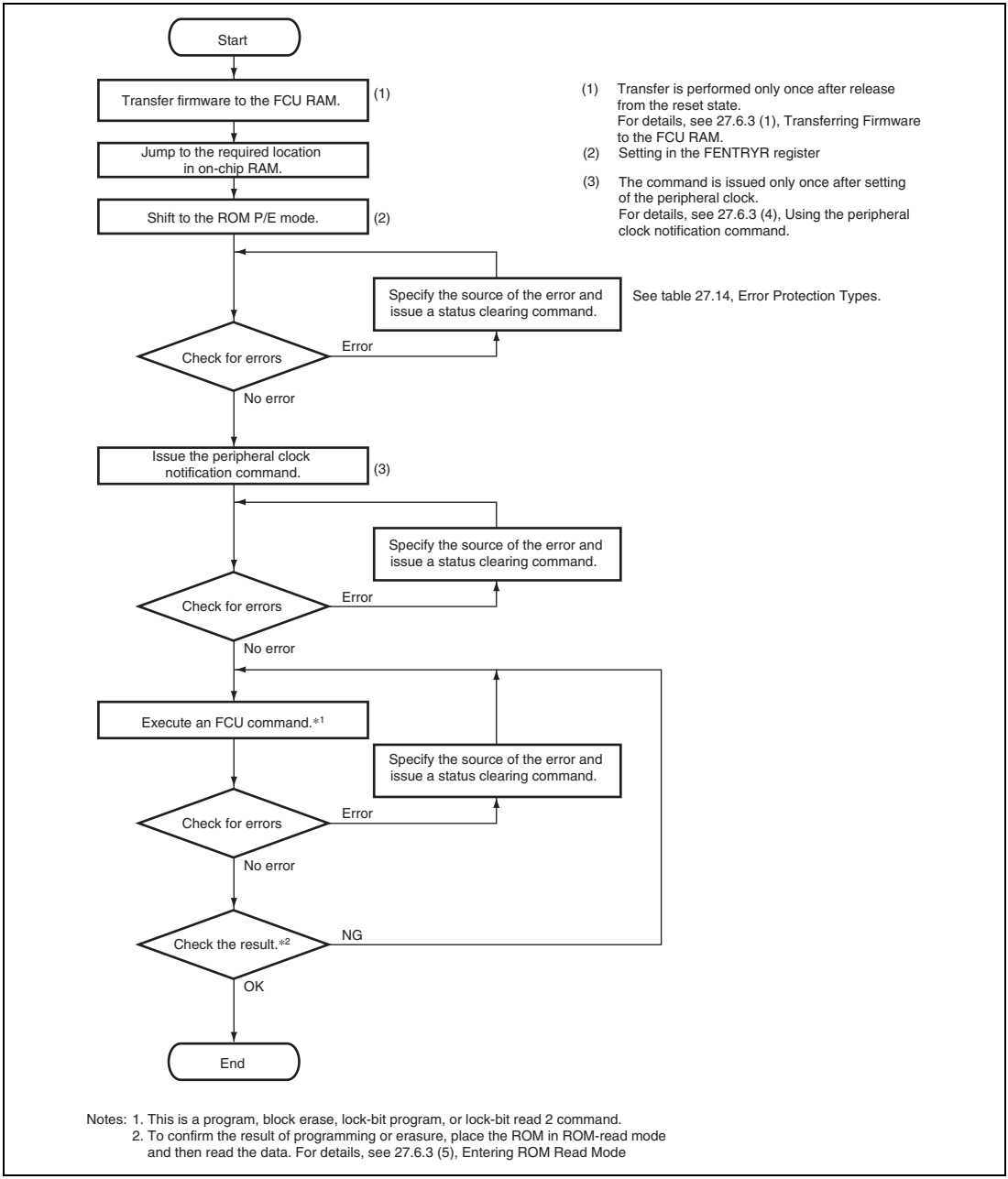
27.6.3 FCU Command Usage

This section shows examples of user processing procedures for firmware transfer to the FCU RAM and the issuing of FCU commands. In some procedures given in this section, the FCU state is not checked before an FCU command is issued but the command result is checked before the processing is completed. To make sure that the FCU accepts a command, check the FCU state before starting processing (see section 27.6.2, Conditions for FCU Command Acceptance).

In a flow used in this section, the current state of FCU command handling and error occurrence is checked via the FRDY, ILGLERR, ERSERR, PRGERR, SUSRDY, ERSSPD, and PRGSPD bits in FSTATR0 and the FCUERR bit in FSTATR1. Since both FSTATR0 and FSTATR1 can be read in word access at a time, the FCU state can be checked by making register access only once. If the FCU state is checked via the FRDY bit of FSTATR0 and the CMDLK bit of FASTAT, register access must be made twice. However, the state of error occurrence can checked via the CMDLK bit only.

The FRDY bit retains 0, if the FRDTCT and FRCRCT bits are set to 1 to put the FCU into a command-locked state in the middle of its command handling while the FCUERR bit is 1. Since the FCU in a command-locked state halts its processes, the FRDY bit is never set to 1 from 0. If the FRDY retains 0 for a longer period than programming/erasing time or suspend delay time (see section 33, Electrical Characteristics), abnormal operation such as the FCU process halt may have occurred. In such case, initialize the FCU by a FCU reset. If the FRDY is set to 1 upon completion of the FCU command handling, the FCUERR bit is also 0. Therefore, the state of error occurrence can be checked via the ILGLERR, ERSERR, and PRGERR bits.

Figure 27.16 gives an overview of the flow of processing for programming and erasure.



Notes: 1. This is a program, block erase, lock-bit program, or lock-bit read 2 command.
2. To confirm the result of programming or erasure, place the ROM in ROM-read mode and then read the data. For details, see 27.6.3 (5), Entering ROM Read Mode

Figure 27.16 Overview of the Flow of Processing for Programming and Erasure

(1) Transferring Firmware to the FCU RAM

To use FCU commands, the FCU firmware must be stored in the FCU RAM. When this LSI is started, the FCU firmware is not stored in the FCU RAM; copy the firmware stored in the FCU firmware area to the FCU RAM. If the FCUERR bit in FSTATR1 is 1, the firmware stored in the FCU RAM may have been damaged; reset the FCU and copy the FCU firmware again in this case.

Figure 27.17 shows the procedure for firmware transfer to the FCU RAM. Before writing data to the FCU RAM, clear FENTRYR to H'0000 to stop the FCU. Transfer firmware to FCU RAM by the CPU or DMAC. For details on the DMAC settings, refer to section 10, Direct Memory Access Controller (DMAC).

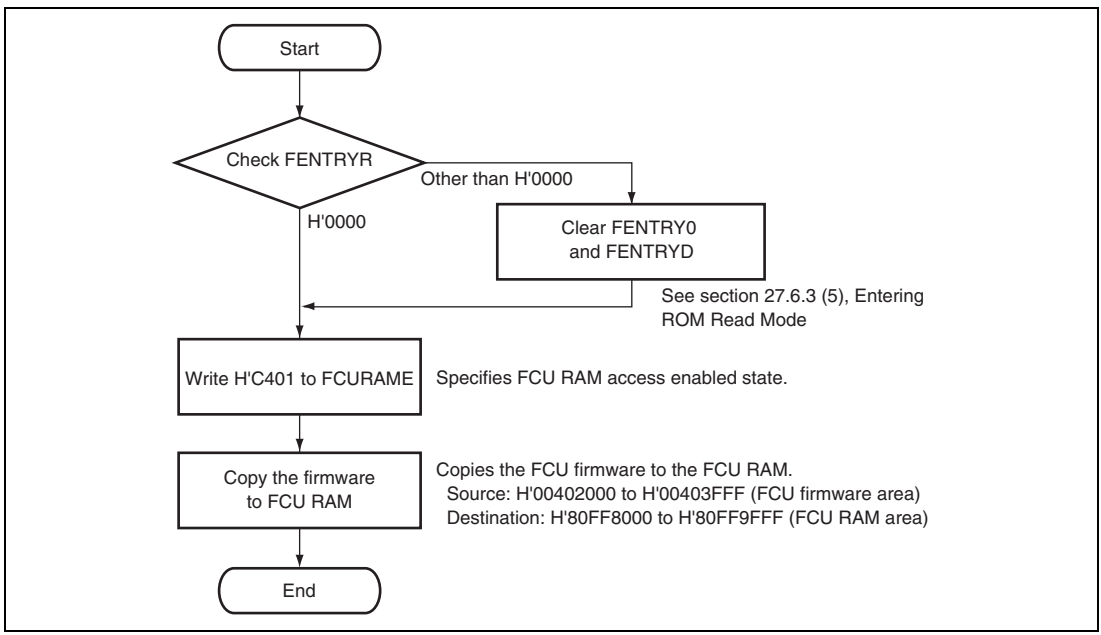


Figure 27.17 Procedure for Firmware Transfer to FCU RAM

(2) Jumping to On-Chip RAM

To prevent the fetching of instructions from the flash memory while it is being programmed or erased, execution must be shifted to an area other than the flash memory (ROM). Copy the required program code to on-chip RAM and then have execution jump to the location of the code in the on-chip RAM.

(3) Entering ROM P/E Mode

To execute ROM-related FCU commands, set the FENTRY0 bit in FENTRYR appropriately to make the FCU enter ROM P/E mode (see section 27.6.2, Conditions for FCU Command Acceptance). For the conditions for writing to the FENTRY0 bit, refer to section 27.3.10, Flash Protect Register (FPROTR).

After a transition from ROM read mode to ROM P/E mode, the FCU is in ROM P/E normal mode.

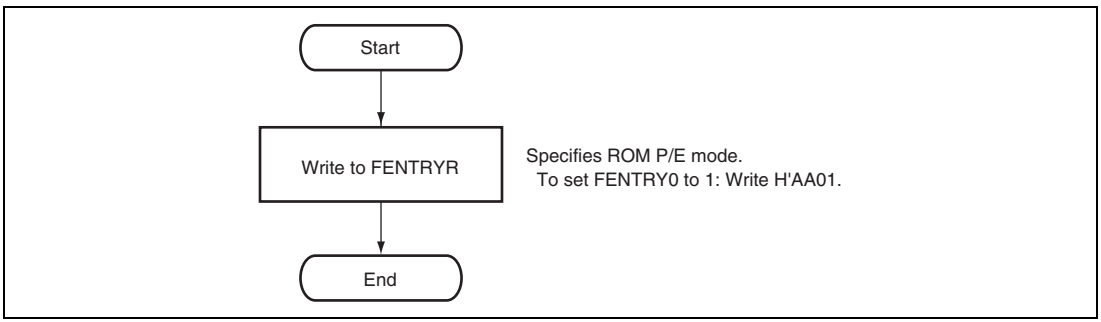


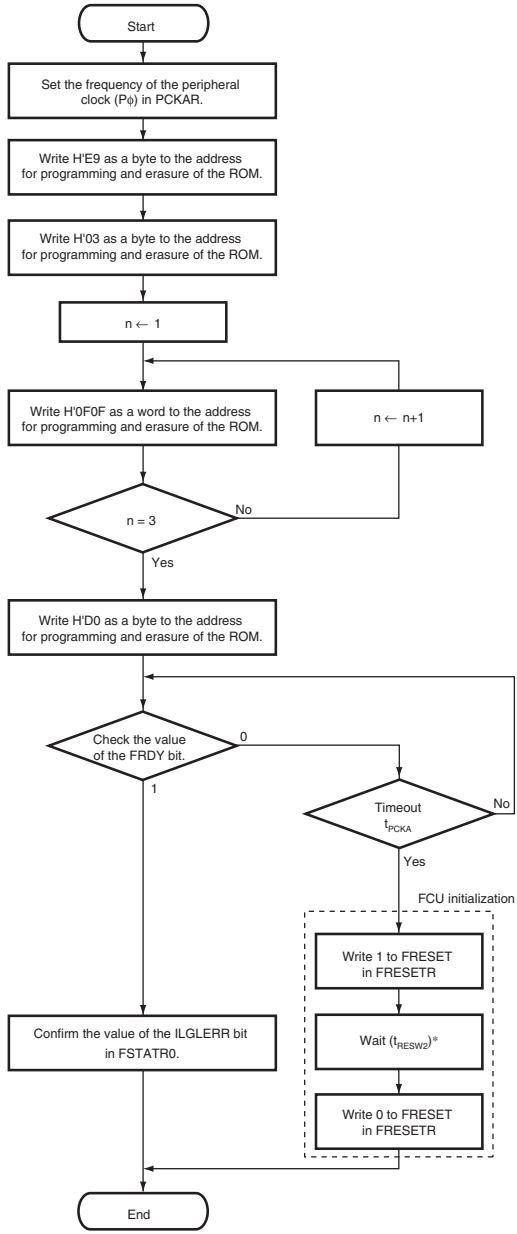
Figure 27.18 Procedure for Transition to ROM P/E Mode

(4) Using the Peripheral Clock Notification Command

The frequency of the peripheral clock to be used before programming or erasure of the flash memory (ROM) must be set in the PCKAR. Selectable values are in the range from 20 to 50 MHz. If the setting is not in this range, the FCU detects an error and enters the command-locked state (see section 27.9.3, Error Protection).

The peripheral clock notification command is used after setting the PCKAR register. For a peripheral clock notification command, H'E9 and H'03 are written in byte units in the first and second cycles, respectively, to the address for programming or erasure of the ROM. In the third to fifth cycles of the command, writing is executed in word units. As the first address, use an address that is aligned with a four-byte boundary. After H'0F0F has been written as a word unit three times to the address for programming or erasure of the ROM, when H'D0 is written as a byte unit to the address for programming or erasure of the ROM, the FCU starts processing for setting the frequency of the peripheral clock. Completion of the setting can be confirmed by checking the value of the FRDY bit in the FSTATR0 register.

After release from the reset state, if the peripheral clock settings in use are not changed, execution once makes the setting valid for subsequent FCU commands.



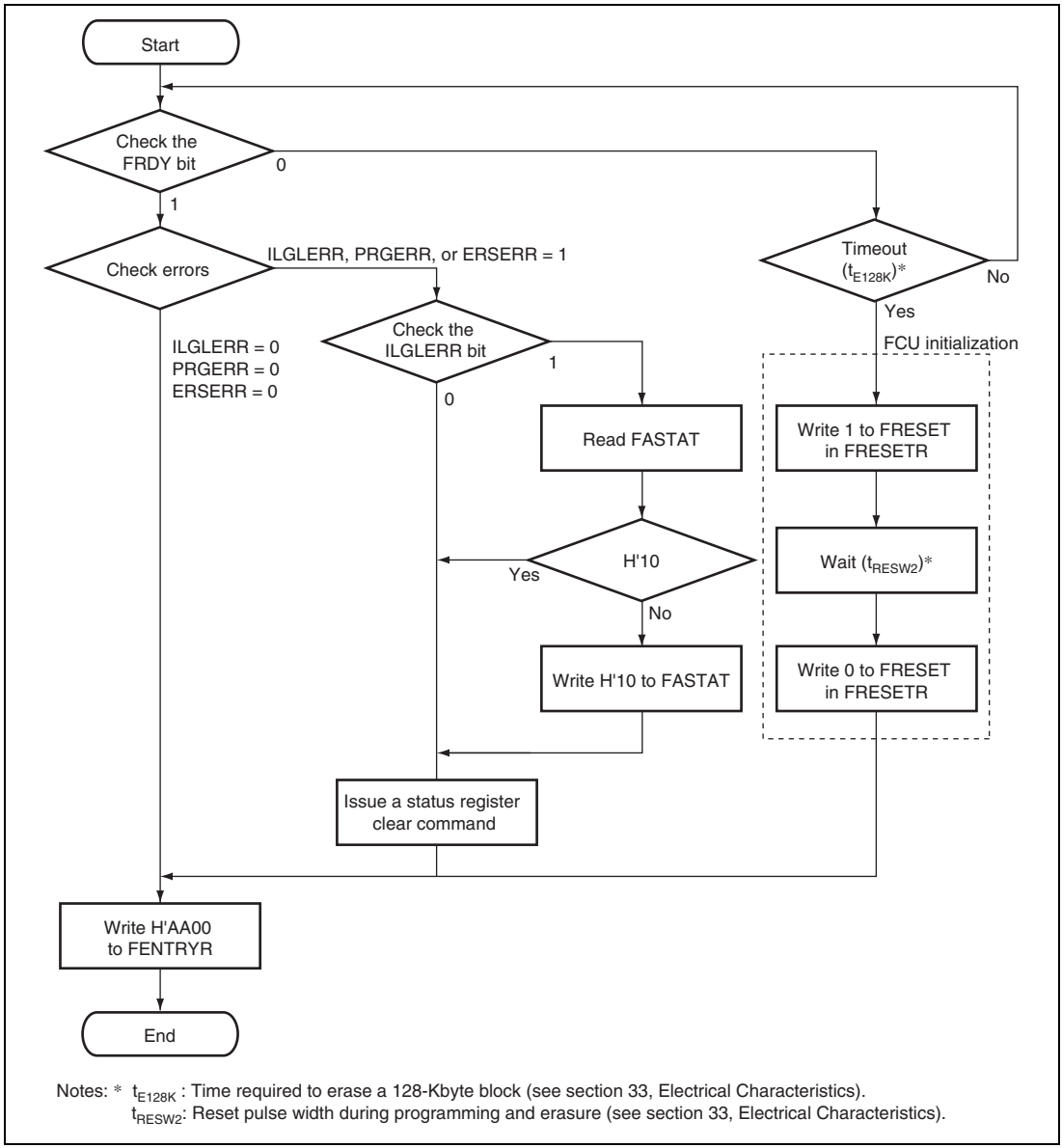
[Legend]
t_{PCKA}: 120 µs for Pϕ = 25 MHz, 60 µs for Pϕ = 50 MHz

Note: * t_{RESW2} denotes the width of a reset pulse during programming or erasure (see section 33, Electrical Characteristics).

Figure 27.19 Flow for Using the Peripheral Clock Notification Command

(5) Entering ROM Read Mode

To enable high-speed ROM read access through the ROM cache, clear the FENTRY0 bit in FENTRYR to make the FCU enter ROM read mode (see section 27.6.2, Conditions for FCU Command Acceptance). A transition from ROM P/E mode to ROM read mode must be made while no FCU error has been detected since FCU command processing is completed.



Notes: * t_{E128K} : Time required to erase a 128-Kbyte block (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 27.20 Procedure for Transition to ROM Read Mode

(6) Using ROM P/E Normal Mode Transition Command

The FCU can be moved to ROM P/E normal mode in two ways: one is to set FENTRYR appropriately in ROM read mode (see section 27.6.3 (1), Transferring Firmware to the FCU RAM) and the other is to issue a normal mode transition command in ROM P/E mode (figure 27.21). The status read mode transition command and the lock bit read mode transition command can be used in the same way as the normal mode transition command.

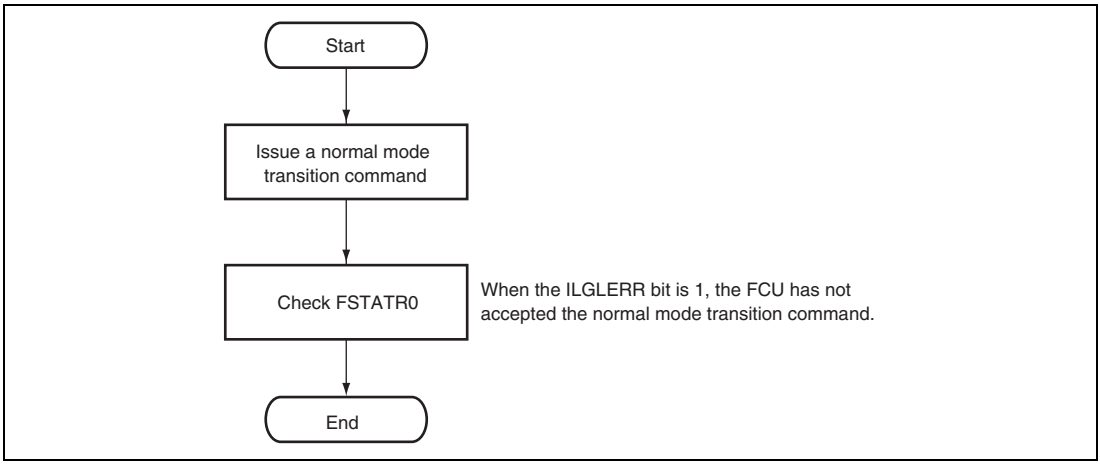
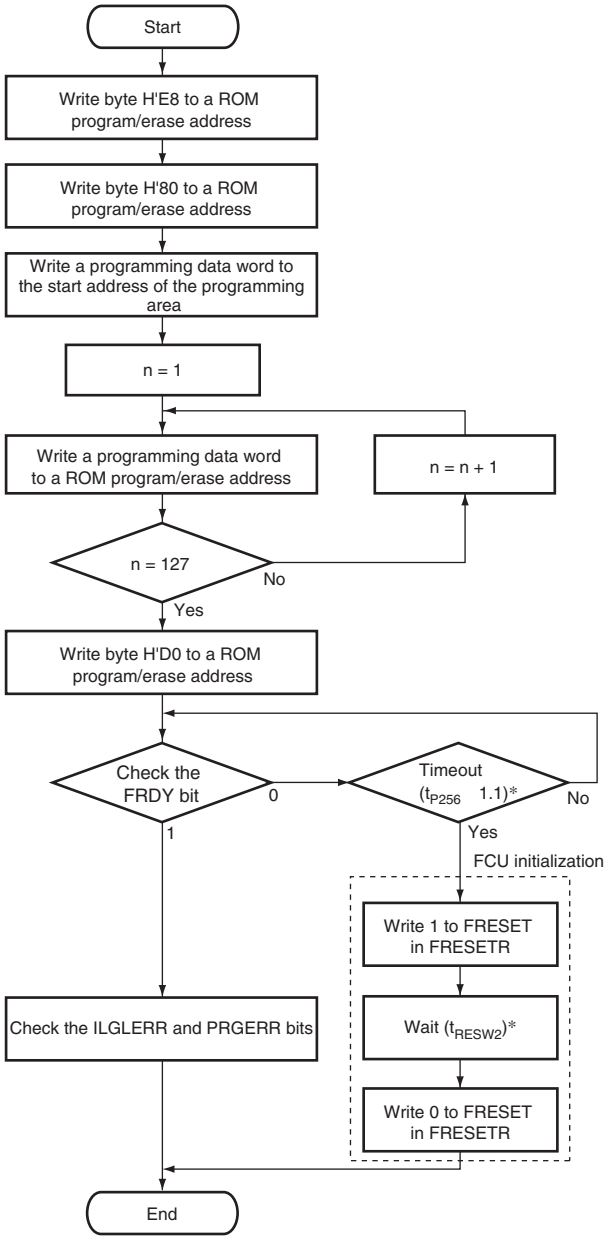


Figure 27.21 Procedure to Use ROM P/E Normal Mode Transition Command

(7) Programming

To program the ROM, use the program command. Write byte H'E8 to a ROM program/erase address in the first cycle of the program command and byte H'80 in the second cycle. Access the P bus in words from the third to 130th cycles of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be a 256-byte boundary address. After writing words to ROM program/erase addresses 127 times, write byte H'D0 to a ROM program/erase address in the 131st cycle; the FCU then starts ROM programming. Read the FRDY bit in FSTATR0 to confirm that ROM programming is completed.

If the area accessed in the third to 130th cycles includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming.



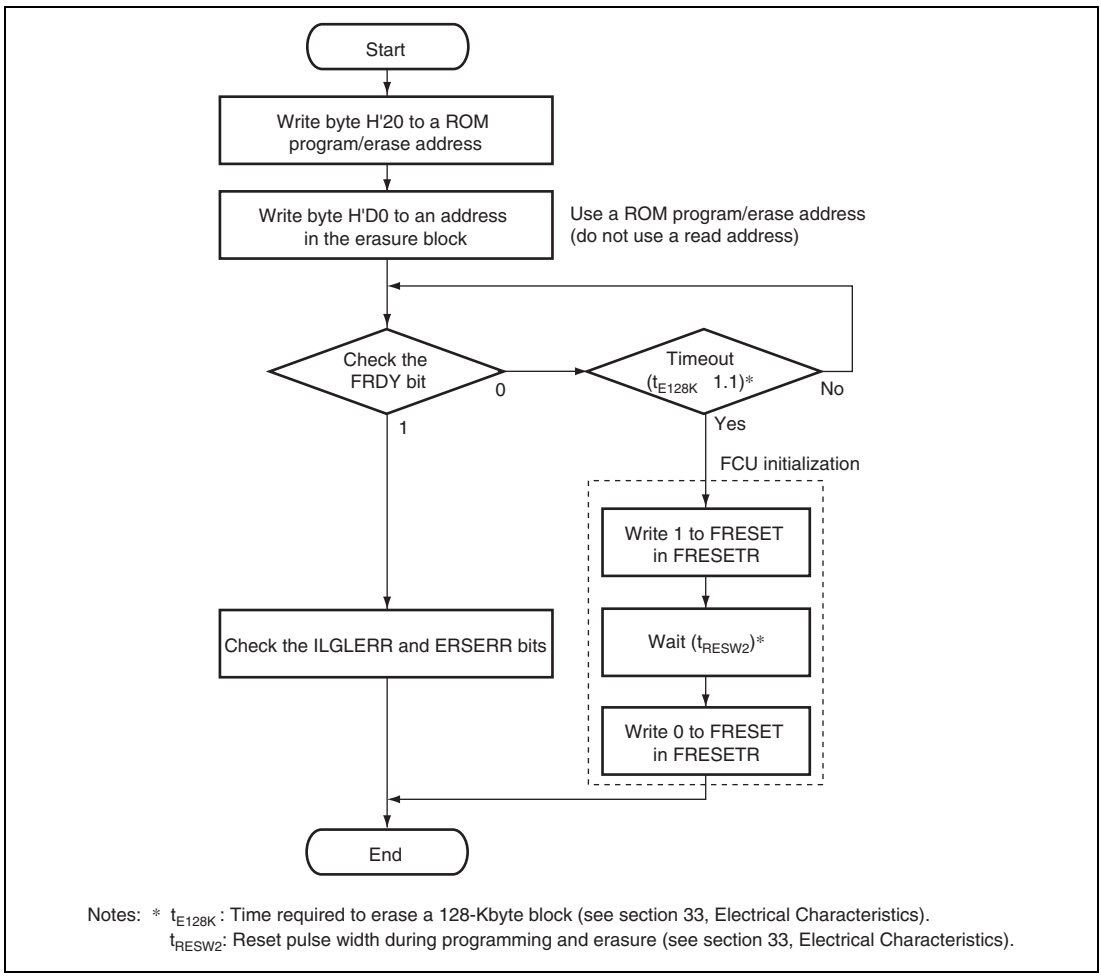
Notes: * t_{P256}: Time required to write 256-byte data (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 27.22 Procedure for ROM Programming

(8) Erasure

To erase the ROM, use the block erase command. Write byte H'20 to a ROM program/erase address in the first cycle of the block erase command. Write byte H'D0 to an address in the target erasure block in the second cycle; the FCU then starts ROM erasure. Read the FRDY bit in FSTATR0 to confirm that ROM erasure is completed.

To ignore the protection provided by the lock bit during erasure, set the FPROTCN bit in FPROTR to 1 before starting erasure.



Notes: * t_{E128K}: Time required to erase a 128-Kbyte block (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

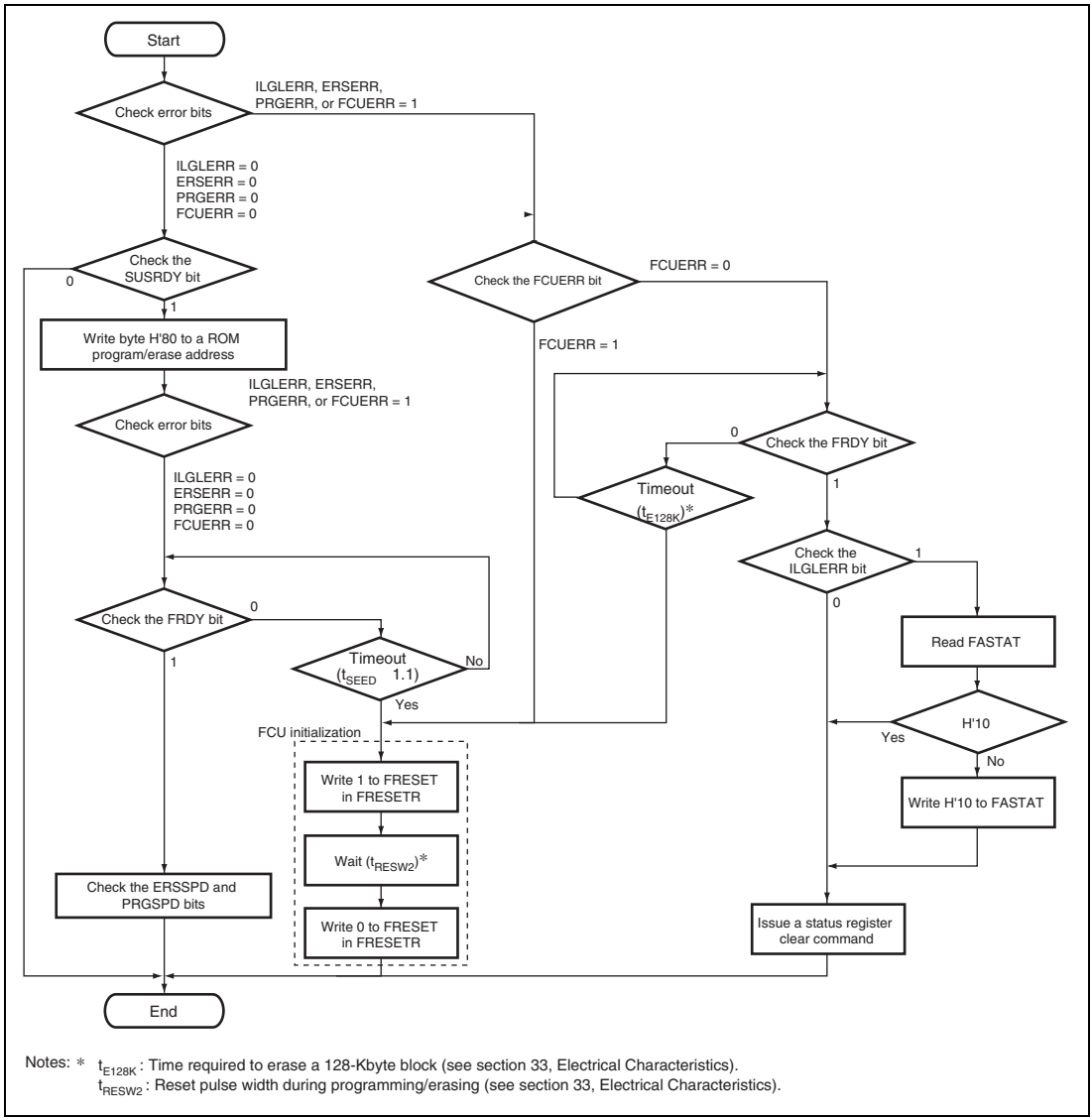
Figure 27.23 Procedure for ROM Erasure

(9) Suspending Programming or Erasure

To suspend programming or erasure of the ROM, use the P/E suspend command. Before issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FCUERR bit in FSTATR1 are 0; that is, to ensure that programming or erasure processing is being performed correctly. Also, check that the SUSRDY bit in FSTATR1 is 1 to ensure that a suspend command is acceptable.

After issuing a P/E suspend command, read both FSTATR0 and FSTATR1 to ensure no error has occurred. If an error has occurred, at least one of the ILGLERR, PRGERR, ERSERR, and FCUERR bits is set to 1. If programming/erasure is complete within the period from when the SUSRDY bit is ensured to be 1 until a P/E suspend command is accepted, the ILGLERR bit is set to 1 as the issued command is detected as illegal. If a P/E suspend command is accepted when programming/erasure is complete, no error occurs, hence no transition to a suspended state (the RDY bit is 1 and both the ERSSPD and PRGSPD bits are 0).

Once a P/E suspend command is accepted and programming/erasure is normally suspended, the FCU enters a suspended state and that the FRDY bit is 1 and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend and ensuring that the FCU has entered a suspend state, determine which operation to perform in the succeeding process. If a P/E resume command is issued in the succeeding process while the FCU has not entered a suspended state, an illegal command error occurs and the FCU enters a command-locked state (see section 27.9.3, Error Protection).



Notes: * t_{E128K} : Time required to erase a 128-Kbyte block (see section 33, Electrical Characteristics).
t_{RESW2} : Reset pulse width during programming/erasing (see section 33, Electrical Characteristics).

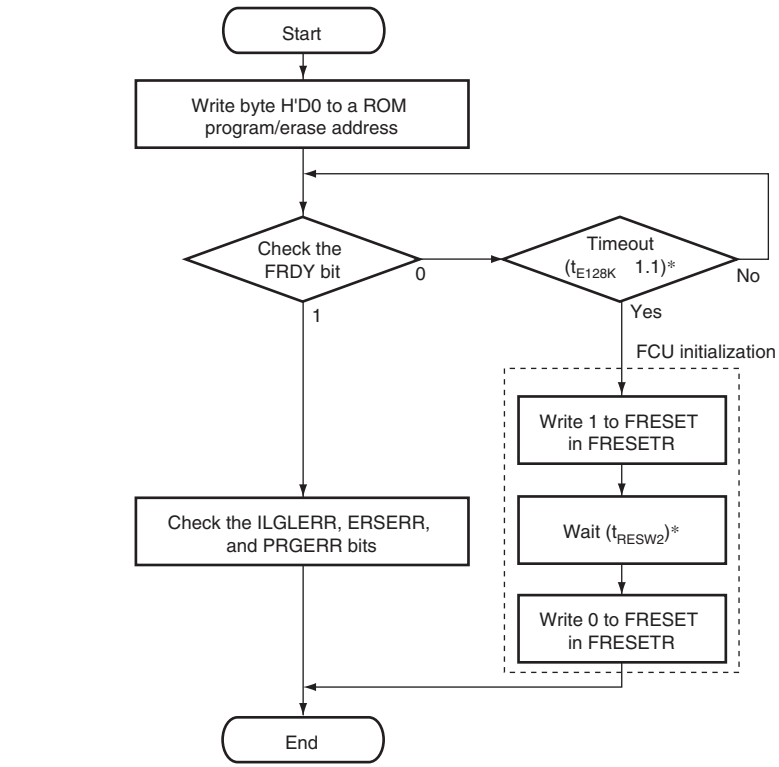
Figure 27.24 Procedure for Programming/Erasure Suspension

Once the FCU has entered the erasure-suspended state, blocks not for erasing can be written to. In both programming-suspended and erasure-suspended states, the FCU can be moved to ROM read mode by clearing FENTRYR.

For the operation when the FCU accepts a P/E suspend command, see section 27.6.4, Suspending Operation.

(10) Resuming Programming or Erasure

To resume programming or erasure that has been suspended, use the P/E resume command. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspension command was issued.



Notes: * t_{E128K} : Time required to erase a 128-Kbyte block (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 27.25 Procedure for Resuming Programming or Erasure

(11) Clearing Status Register 0 (FSTATR0)

To clear the ILGLERR, PRGERR, and ERSERR bits in FSTATR0, use the status register clear command. When any one of the ILGLERR, PRGER, and ERSERR bits is 1, the FCU is in command-locked state, in which the FCU only accepts the status register clear command and does not accept other commands. When the ILGLERR bit is 1, check also the value of the ROMAE, EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTAT. If a status register clear command is issued without clearing these bits, the ILGLERR bit is not cleared.

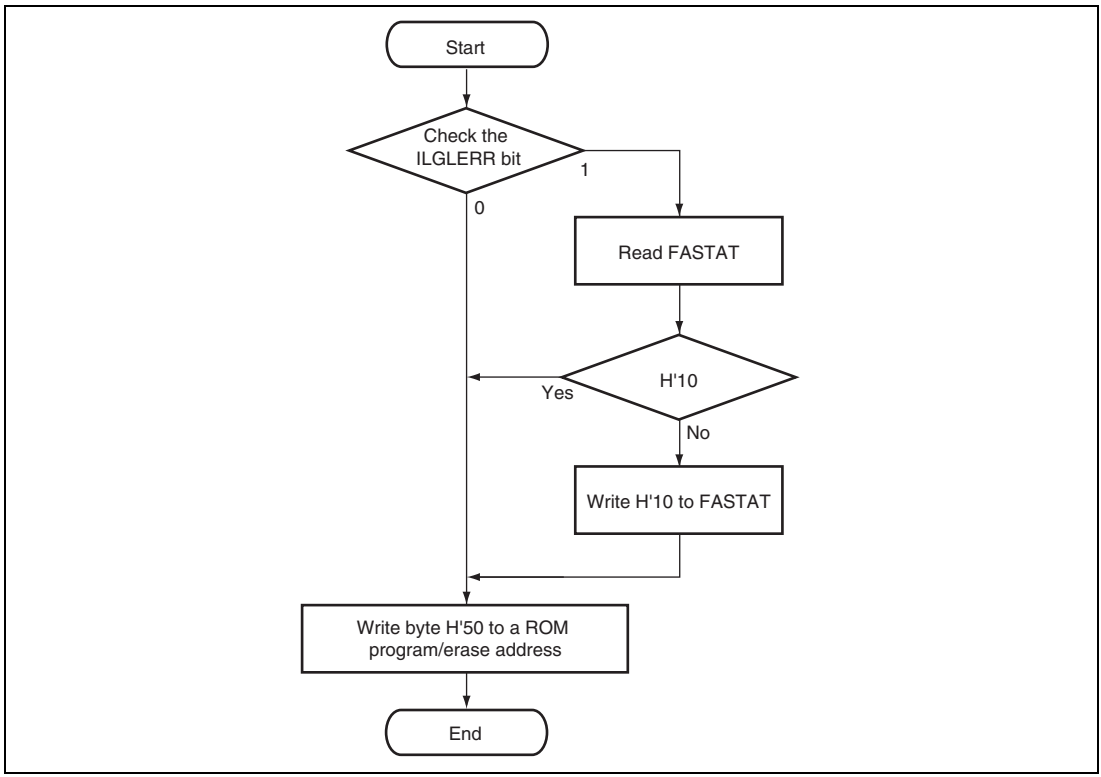


Figure 27.26 Procedure for Clearing Status Register 0

(12) Checking Status Register 0 (FSTATR0)

The FSTATR0 value can be checked in two ways: one is to directly read FSTATR0 and the other is to read a ROM program/erase address in ROM status read mode. After an FCU command is issued that is neither a normal mode transition command nor a lock bit read mode transition command, the FCU is in ROM status read mode. In the example shown in figure 27.27, a status read mode transition command is issued to enter ROM status read mode, and then a ROM program/erase address is read to check the FSTATR0 value.

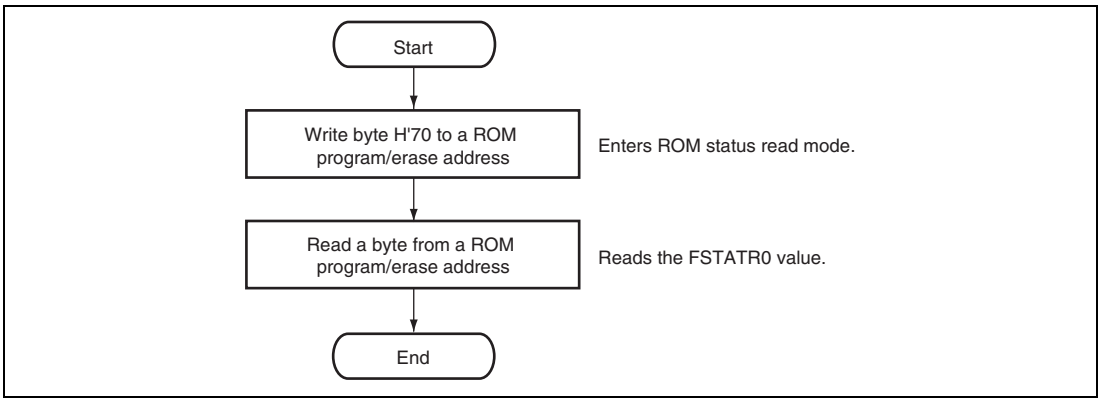


Figure 27.27 Procedure for Checking Status Register 0

(13) Reading Lock Bit

Each erasure block in the user MAT has a lock bit. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased.

The lock bit status can be checked in either memory area read mode or register read mode. In memory area read mode (the FRDMD bit in FMODR is 0), read a ROM program/erase address in ROM lock bit read mode, and the lock bit value in the specified erasure block is copied to all bits in the data read through the P bus. In register read mode (the FRDMD bit in FMODR is 1), issue a lock bit read 2 command, and the lock bit value in the specified erasure block is copied to the FLOCKST bit in FSTATR1.

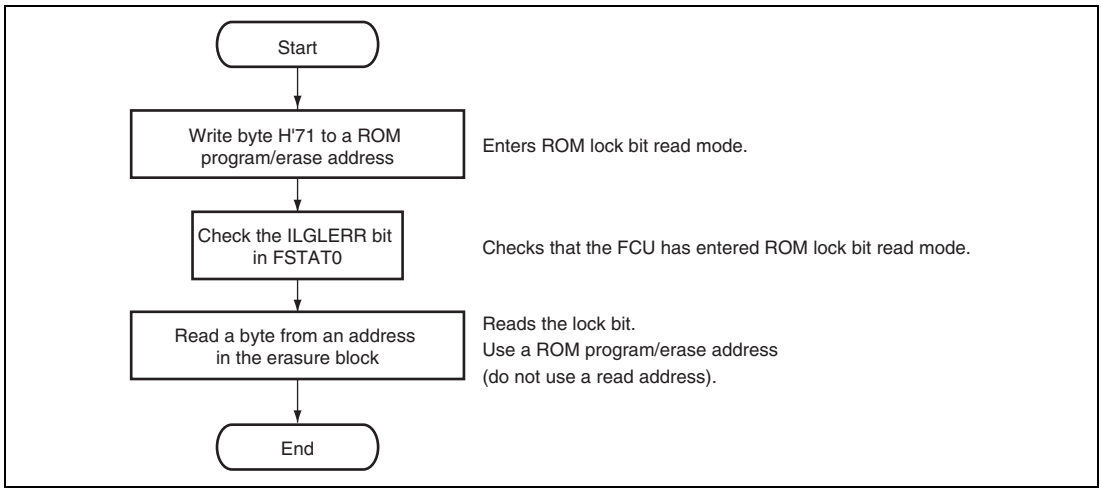
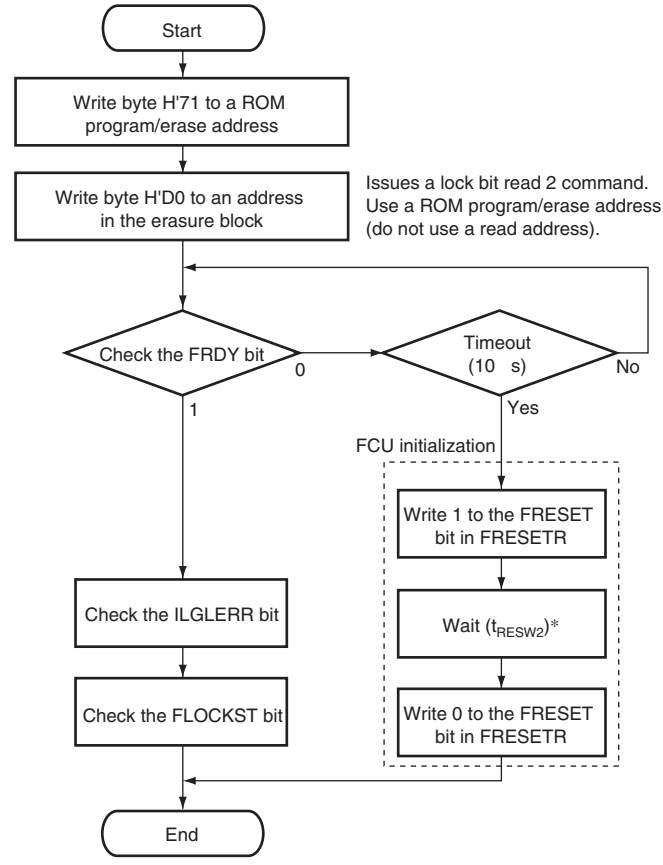


Figure 27.28 Procedure for Reading Lock Bit in Memory Area Read Mode

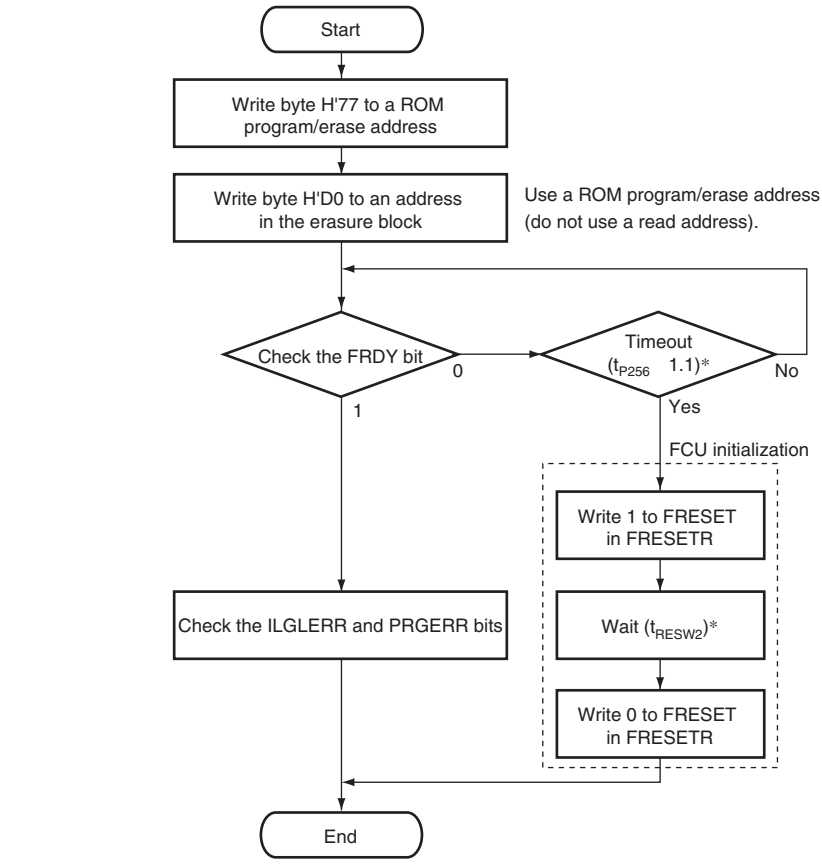


Note: * t_{RESW2} : Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 27.29 Procedure for Reading Lock Bit in Register Read Mode

(14) Writing to Lock Bit

Each erasure block in the user MAT has a lock bit. To write to a lock bit, use the lock bit program command. Write byte H'77 to a ROM program/erase address in the first cycle of the lock bit program command. Write byte H'D0 to an address in the target erasure block whose lock bit is to be written to in the second cycle; the FCU then starts writing to the lock bit. Read the FRDY bit in FSTATR0 to confirm that writing is completed.



Notes: * t_{P256}: Time required to write 256-byte data (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 27.30 Procedure for Writing to the Lock Bit

To erase a lock bit, use the block erase command. While the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be erased. Set the FPROTCN bit to 1, and then issue a block erase command to erase a lock bit. The block erase command erases all data in the specified erasure block; it is not possible to erase only the lock bit.

27.6.4 Suspending Operation

When a P/E suspend command is issued while ROM is being programmed or erased, the FCU suspends the programming or erasure processing. Figure 27.31 gives an overview of operation for suspending programming. Upon accepting a programming command, the FCU clears the FRDY bit in FSTATR0 to 0 and starts programming. Once the FCU enters a state where it is ready to accept a command after the start of programming, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command while reapplying a write pulse, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse, suspends programming, and sets the PRGSPD bit to 1. Once the process completes, the FCU sets the FRDY bit to 1 and enters a programming suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts programming.

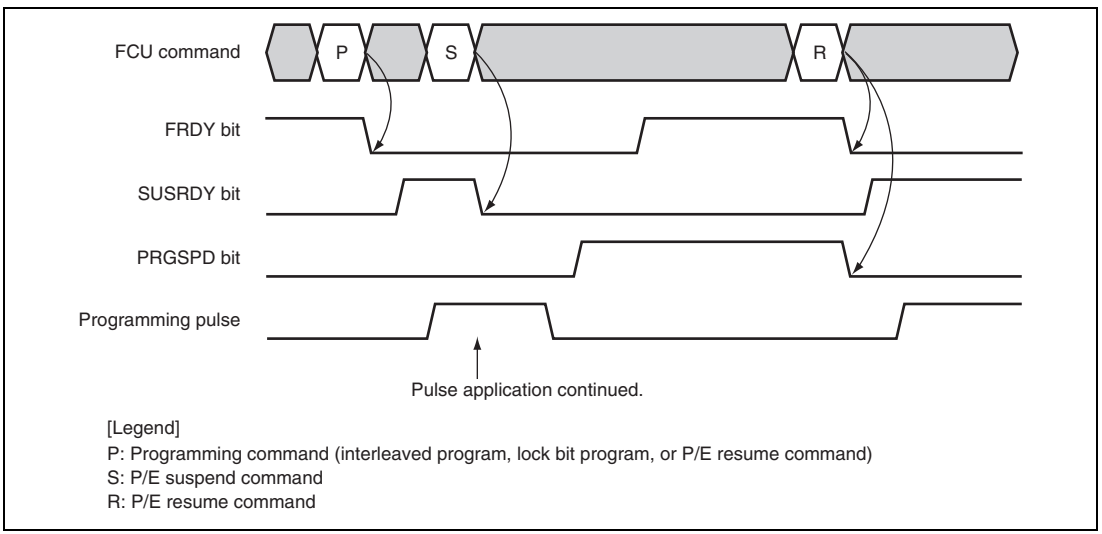


Figure 27.31 Suspending Programming Processing

Figure 27.32 shows the operation for suspending erasure processing in suspension-priority mode (the ESUSPMD bit in FCPSR is 0). Upon accepting an erasing command, the FCU clears the FRDY bit to 0 and starts erasing. Once the FCU enters a state where it is ready to accept a command after the start of erasing, the SUSRDY bit is set to 1. If a P/E suspend command is issued, the FCU accepts the command and clears the SUSRDY bit. If the FCU accepts the command during its erasing operation, the FCU starts a suspending process even while applying a pulse and sets the ERSSPD bit to 1. Once the suspending process completes, the FCU sets the FRDY bit to 1 and enters an erasing suspended state. If the FCU accepts a P/E resume command in this state, the FCU clears the FRDY and PRGSPD bits to 0 and restarts erasing. The operations of the FRDY, SUSRDY, and ERSSPD bits are independent of the erasure-suspended mode.

The setting for the erasure-suspended mode affects the control methods for erasure pulse. In suspend-priority mode, if the FCU accepts a P/E suspend command while applying erasure pulse A, which has not been suspended previously, the FCU suspends the pulse application and enters an erasure-suspended state. After the FCU resumes erasing by accepting a P/E resume command, if the FCU accepts a P/E suspend command while applying erasing pulse A, the FCU continues applying the pulse. After a specified pulse application time has elapsed, the FCU completes applying the pulse and enters an erasure-suspended state. Next, after the FCU accepts a P/E resume command and starts applying a new pulse B, if the FCU accepts a P/E suspend command, the FCU suspends the pulse application. In suspense-priority mode, the suspense process is given priority by suspending once every pulse application.

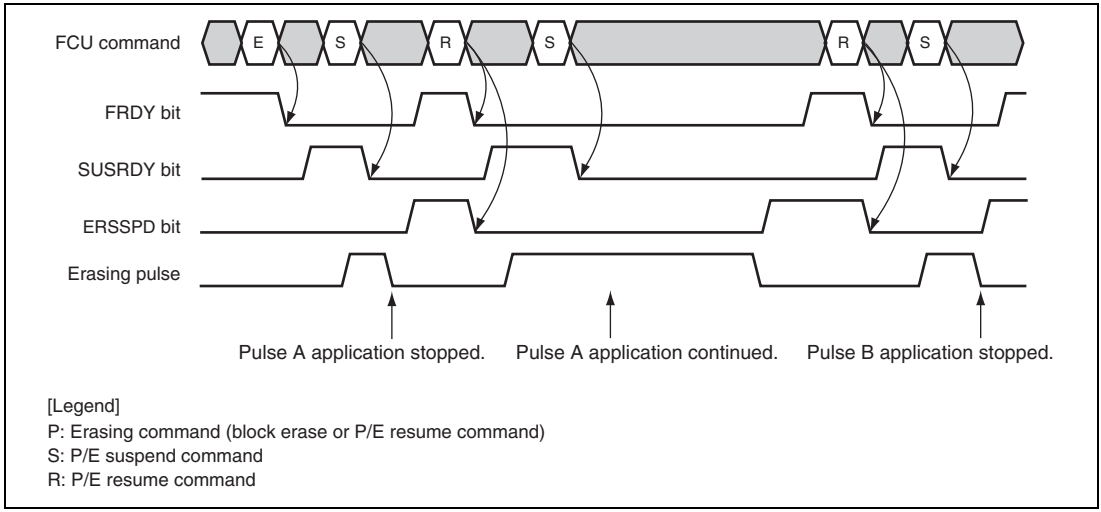


Figure 27.32 Suspending Erasure Processing (Suspension-Priority Mode)

Figure 27.33 shows how erasure processing is suspended in erasure-priority mode (with the ESUSPMD bit in FCPSR being 1). The operation for suspending erasure processing in erasure-priority mode (the ESUSPMD bit in FCPSR is 1) is equivalent to that for suspending programming processing.

In erasure-priority mode, if the FCU accepts a P/E suspend command while applying an erasing pulse, the FCU always continues applying the pulse. As processing to reapply an erasing pulse never takes place in this mode, the total time required for erasure processing is shorter than in suspension-priority mode.

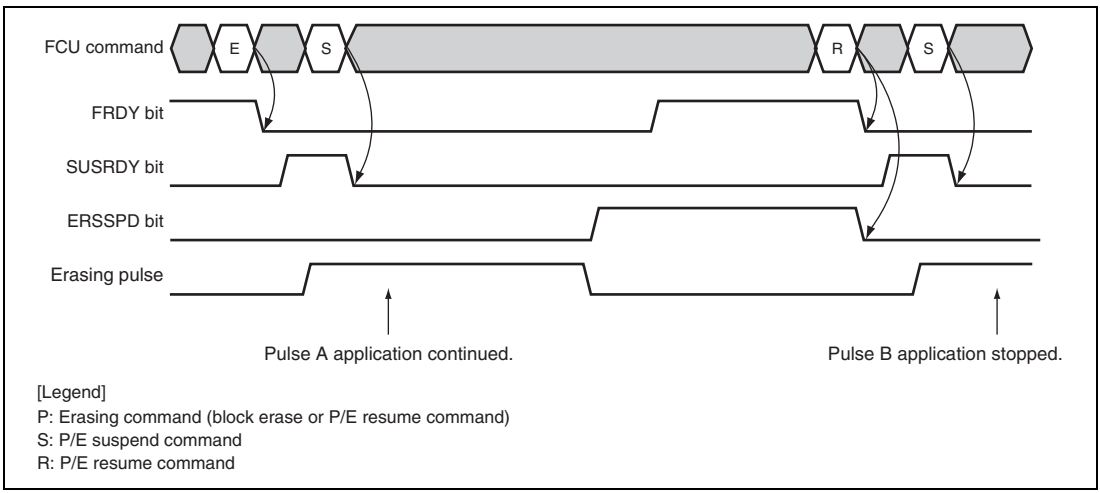


Figure 27.33 Suspending Erasure Processing (Erasure-Priority Mode)

27.7 User Boot Mode

To program or erase the user MAT in user boot mode, issue FCU commands to the FCU. A user-defined boot mode can be implemented by writing to the user boot MAT a ROM programming/erasing routine that uses a desired communications interface; when this LSI is started in user boot mode after that, the user-defined boot mode is initiated. Programming/erasure of the user boot MAT is only enabled in boot mode.

27.7.1 User Boot Mode Initiation

When this LSI is started in user boot mode, execution starts in the embedded program stored MAT, necessary processing such as FCU firmware transfer to the FCU RAM is performed, and then execution jumps to the location indicated by the reset vector of the user boot MAT. Figure 27.34 gives an overview of the boot sequence.

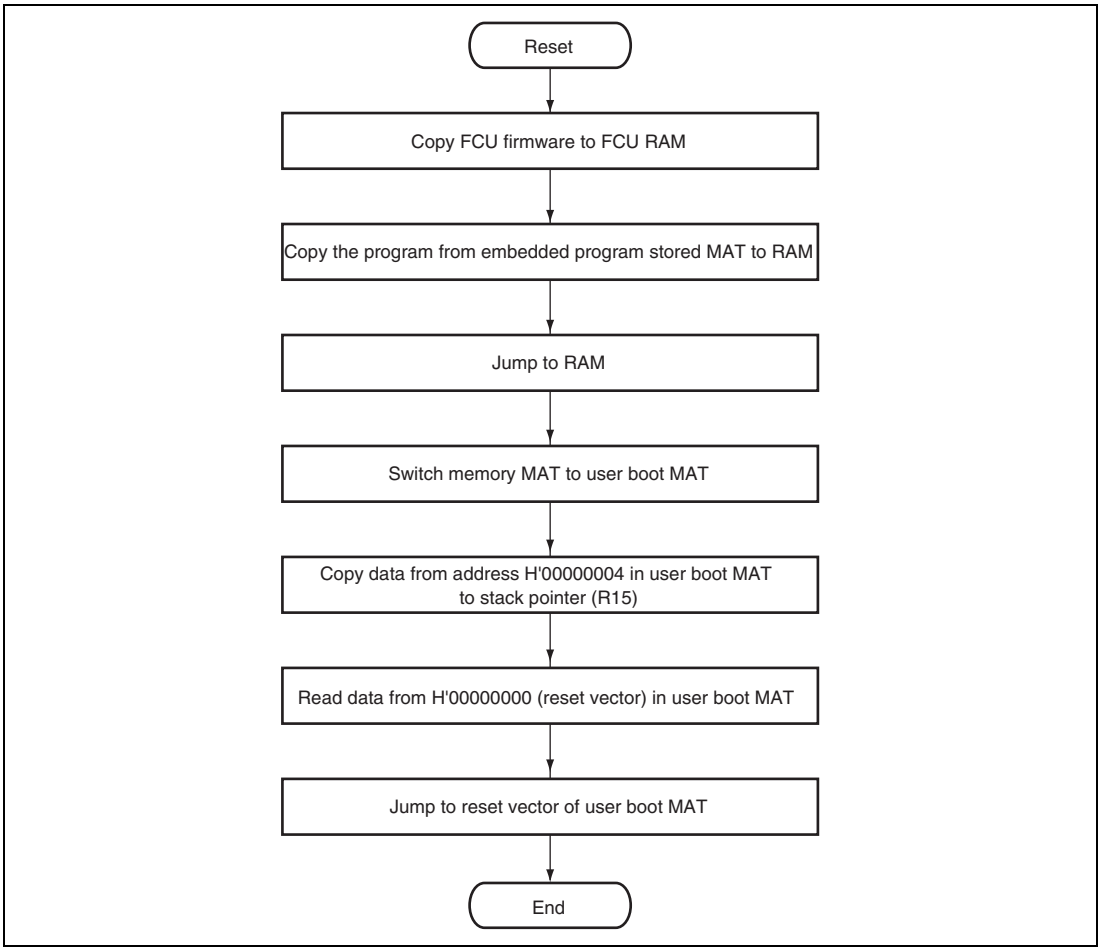


Figure 27.34 Overview of Boot Sequence in User Boot Mode

27.7.2 User MAT Programming

The user MAT can be programmed by starting this LSI in user boot mode while the user MAT programming/erasing routine created by the user is stored in the user boot MAT. Be sure to copy the user MAT programming/erasing routine to the RAM and execute it in the RAM. The user boot MAT is selected in the initial state in user boot mode; be sure to switch the memory MAT to the user MAT before starting programming. If an FCU command for ROM programming or erasure is issued while the user boot MAT is selected, the FCU does not program or erase the ROM. Figure 27.35 shows an example of the user MAT programming procedure.

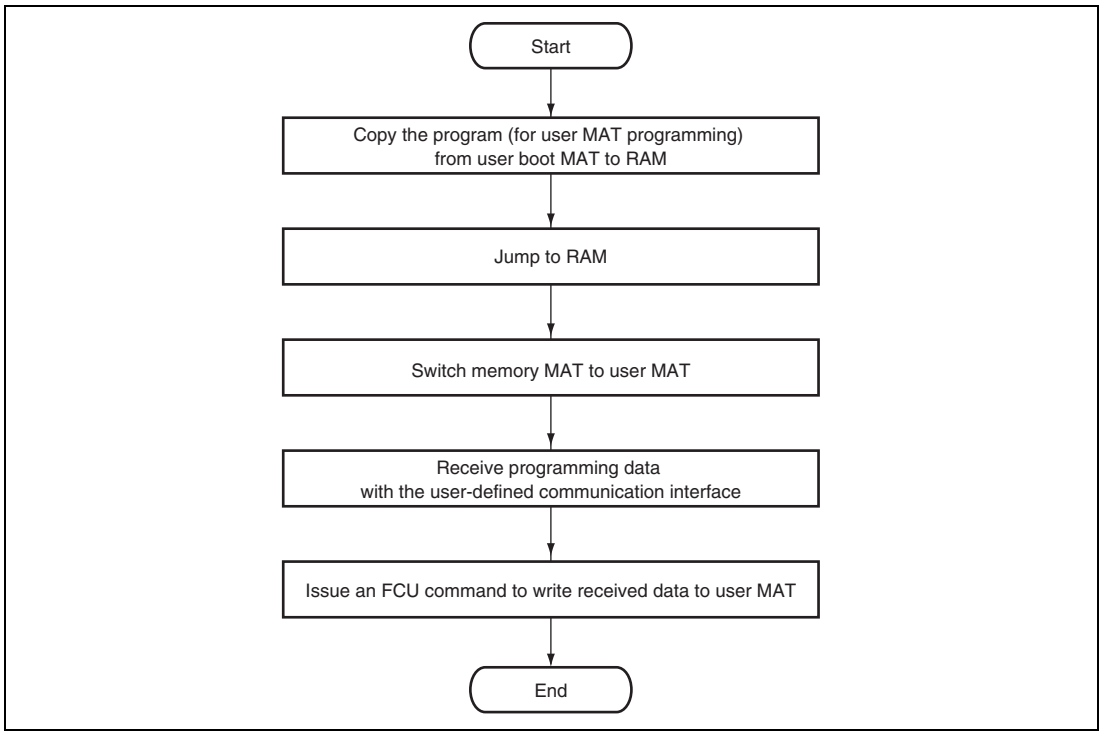


Figure 27.35 Example of User MAT Programming

27.8 Programmer Mode

In programmer mode, a PROM programmer can be used to perform programming/erasing via a socket adapter, just as for a discrete flash memory. Use a PROM programmer that supports the MCU device type (FZTAT1024DV3A) having on-chip Renesas 1-Mbyte flash memory.

27.9 Protection

There are three types of ROM programming/erasure protection: hardware, software, and error protection.

27.9.1 Hardware Protection

The hardware protection function disables ROM programming and erasure according to the LSI pin settings.

(1) Protection through FWE Pin

When a low level is applied to the FWE pin, the FWE bit in FPMON becomes 0. In this state, a 1 cannot be written to the FENTRY0 bit in FENTRYR; that is, ROM P/E mode cannot be entered, which prevents the ROM from being programmed or erased.

When the FRDY bit is 1 and the FWE pin is driven low, the FCU clears the FENTRY0 bit to disable ROM programming and erasure. If the FRDY bit in FSTATR0 has already been set to 0 before the FWE pin is driven low, the FCU continues command processing. Even while processing a command, the FCU can accept a P/E suspend command. To resume programming or erasing the ROM, reset the FENTRY0 bit to the value that was set before being cleared, and then issue a P/E resume command.

If an attempt is made to issue a programming or erasing command to the ROM against the protection through the FWE pin, the FCU detects an error and enters command-locked state.

(2) Protection through Mode Pins

While the on-chip ROM is disabled, ROM programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 3, MCU Operating Modes. In user boot mode or user program mode, the user boot MAT cannot be programmed or erased.

27.9.2 Software Protection

The software protection function disables ROM programming and erasure according to the control register settings or the lock bit settings in the user MAT. If an attempt is made to issue a programming or erasing command to the ROM against software protection, the FCU detects an error and enters command-locked state.

(1) Protection through FENTRYR

When the FENTRY0 bit is 0, the 1-Mbyte ROM (read addresses: H'00000000 to H'000FFFFF; program/erase addresses: H'80800000 to H'808FFFFF) is set to ROM read mode. In ROM read mode In ROM read mode, the FCU does not accept commands, so ROM programming and erasure are disabled. If an attempt is made to issue an FCU command in ROM read mode, the FCU detects an illegal command error and enters command-locked state (see section 27.9.3, Error Protection).

(2) Protection through Lock Bits

Each erasure block in the user MAT has a lock bit. When the FPROTCN bit in FPROTR is 0, the erasure block whose lock bit is set to 0 cannot be programmed or erased. To program or erase the erasure block whose lock bit is 0, set the FPROTCN bit to 1. If an attempt is made to issue a programming or erasing command against protection by lock bits, the FCU detects an programming/erasure error and enters command-locked state (see section 27.9.3, Error Protection).

27.9.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the ROM cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While the ROMAEINT bit in FAEINT is 1, an FIFE interrupt is generated if the ROMAE bit in FASTAT becomes 1.

Table 27.14 shows the error protection types dedicated for the ROM, those used in common by the ROM and the FLD, and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FCUERR bit in FSTATR1, and the ROMAE bit in FASTST) after each error detection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the ROM. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

Table 27.14 Error Protection Types

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
FENTRYR setting error	The value set in FENTRYR is not H'0001, H'0002, H'0008, H'0010, or H'0080.	1	0	0	0	0
	The FENTRYR setting for resuming operation does not match that for suspending operation.	1	0	0	0	0

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
Illegal command error	An undefined code has been specified in the first cycle of an FCU command.	1	0	0	0	0
	The value specified in the last of the multiple cycles of an FCU command is not H'D0.	1	0	0	0	0
	The peripheral clock specified in PCKAR is not in the range from 1 to 100 MHz.	1	0	0	0	0
	The command issued during programming or erasure is not a suspend command.	1	0	0	0	0
	A suspend command has been issued during operation that is neither programming nor erasure.	1	0	0	0	0
	A suspend command has been issued in suspended state.	1	0	0	0	0
	A resume command has been issued in a state that is not a suspended state.	1	0	0	0	0
	A programming or erasing command (program, lock bit program, block erase) has been issued in programming-suspended state.	1	0	0	0	0
	A block erase command has been issued in erasure-suspended state.	1	0	0	0	0
	A program, lock bit program, or non-interleaved program command has been issued for an erasure-suspended area in erasure-suspended state.	1	0	0	0	0
	The value specified in the second cycle of a program command is not H'80.	1	0	0	0	0
	A command has been issued in command-locked state.	1	0/1	0/1	0/1	0/1
Erasure error	An error has occurred during erasure processing.	0	1	0	0	0
	A block erase command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.	0	1	0	0	0
Programming error	An error has occurred during programming processing.	0	0	1	0	0
	A program, lock bit program, or program command has been issued for the erasure block whose lock bit is set to 0 while the FPROTCN bit in FPROTR is 0.	0	0	1	0	0

Error	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE
FCU error	An error has occurred during CPU processing in the FCU.	0	0	0	1	0
ROM access error	A read access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 1 in ROM P/E normal mode.	1	0	0	0	1
	An access command has been issued to addresses H'80800000 to H'808FFFFF while FENTRY0 = 0	1	0	0	0	1
	A read access command has been issued to addresses H'00000000 to H'001FFFFF while the FENTRYR register value is not H'0000	1	0	0	0	1
	A ROM programming or erasing command (interleaved program, lock bit program, or block erase command) has been issued while the user boot MAT is selected.	1	0	0	0	1
	An access command has been issued to an address other than the addresses for ROM programming/erasure H'80800000 to H'80807FFF while the user boot MAT is selected.	1	0	0	0	1

27.10 Usage Notes

27.10.1 Switching between User MAT and User Boot MAT

The user MAT and user boot MAT are allocated to the same address area. If the ROM area is accessed during switching between the user MAT and user boot MAT, an unexpected MAT may be accessed because the number of cycles required to access the ROM area depends on the internal bus status. When the ROM cache function is enabled, the previously stored data is left in the ROM cache even after MAT switching; note that a cache hit may occur when a newly selected MAT is accessed at the same address as the data stored in the cache. To avoid such unexpected behavior, take the following steps before and after MAT switching.

1. Modifying interrupt settings before MAT switching

   There are two ways to avoid ROM area access due to an interrupt during MAT switching: one is to specify the interrupt vector fetch destination outside the ROM area through the vector base register (VBR) setting in the CPU, and the other is to mask interrupts. Note that NMI interrupts cannot be masked in this LSI; when masking interrupts to avoid ROM area access in this LSI, design the system so that no NMI is generated during MAT switching.
2. Switching between MATs through a program outside the ROM area

   To avoid CPU instruction fetch in the ROM area during MAT switching, execute the MAT switching processing outside the ROM area.
3. Performing dummy read of ROMMAT

   After writing to ROMMAT to switch between MATs, perform a dummy read of ROMMAT to ensure that the register write is completed.
4. Flushing the ROM cache after MAT switching

   Disable (flush) the instructions or data in the ROM cache by writing a 1 to the RCF bit in RCCR.

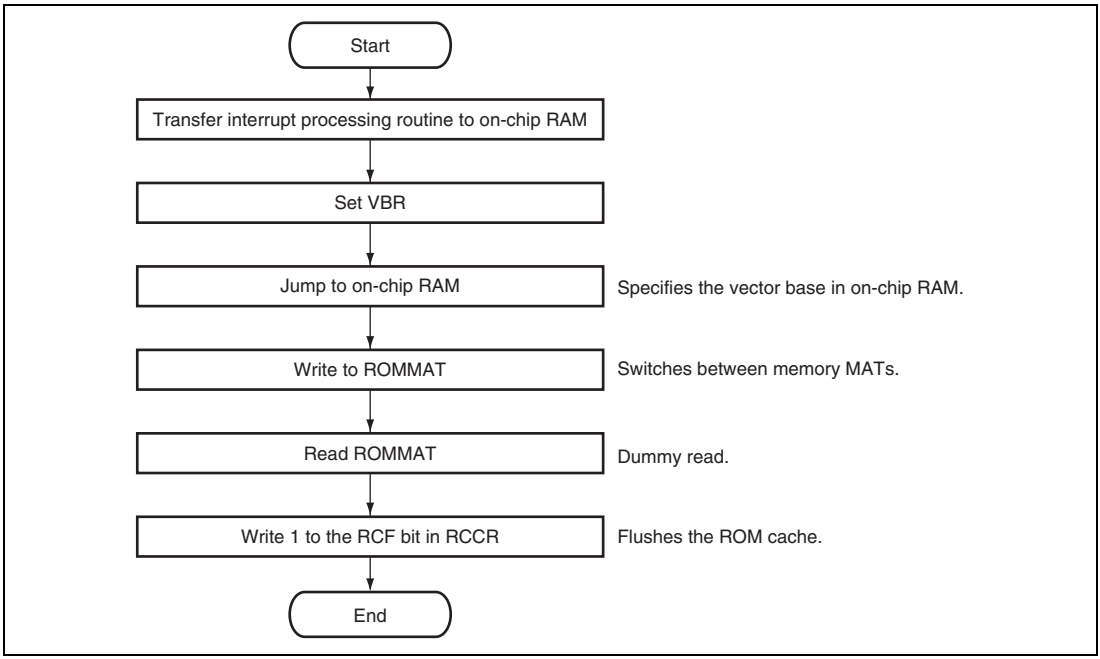


Figure 27.36 Example of MAT Switching Steps

27.10.2 State in which Interrupts are Ignored

In the following mode or period, the AUD is in module standby mode and cannot operate. The NMI or maskable interrupt requests are ignored.

- Boot mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

27.10.3 Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no instruction is executed or no data is read from the programming-suspended or erasure-suspended area.

To avoid instruction fetch from the programming-suspended or erasure-suspended area, which may be caused by prefetch by the ROM cache, ensure that no instruction is fetched within 16 bytes from the start address of the programming-suspended or erasure-suspended area.

During ROM cache prefetch, the destination of a branch instruction is also accessed. The destination must not be in the programming-suspended or erasure-suspended area.

27.10.4 Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcomputers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

27.10.5 FWE Pin State

Ensure that the FWE pin level does not change during programming or erasure. If the FWE level goes low, the current programming or erasure terminates abnormally and the FRDY bit is set to 1 (the erasure or programming error bit in FASTATR0 is set), and then FENTRYR is cleared. To reprogram ROM, do it after erasing data with the FWE pin at the high level.

In a transition from single-chip mode to user program mode, issue an FCU command after driving the FWE pin high, making sure that the FWE bit in FPMON is set to 1, and setting the FENTRYR register.

In a transition from user program mode to single-chip mode, drive the FWE pin low after ROM programming is completed, making sure that the FRDY bit in FSTATR0 is set to 1, and clearing the FENTRYR register.

For ROM protection in a mode that begins with the FWE pin at the high level, drive the FWE pin low tMDH1 after the reset is cleared.

Cancel ROM protection using the same steps as the transition from single-chip mode to user program mode, and set ROM protection using the same steps as the transition from user program mode to single-chip mode.

27.10.6 Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t_{RESW2} (see section 33, Electrical Characteristics). Since a high voltage is applied to the ROM during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the ROM while the FCU is in the reset state.

When a power-on reset is generated by asserting the $\overline{\text{RES}}$ pin during programming or erasure of the flash memory, hold the reset state for a period of t_{RESW2} (see section 33, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the ROM and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

When executing a power-on reset by asserting the $\overline{\text{RES}}$ pin or the FCU reset with the FRESET bit set in FRESETR during programming/erasure, all data including a lock bit of a programming/erasure target area are undefined.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the ROM, or initialization of its internal circuit.

27.10.7 Suspension by Programming/Erasure Suspension

When suspending programming/erasure processing with the programming/erasure suspend command, make sure to complete the operations with the resume command.

27.10.8 Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

27.10.9 Allocation of Interrupt Vectors during Programming and Erasure

Generation of an interrupt during programming and erasure can lead to fetching from the vector in the flash memory (ROM). For this reason, prepare the interrupt vector table and the interrupt processing routines in areas other than the flash memory (ROM).

27.10.10 Items Prohibited during Programming and Erasure

High voltages are applied within the flash memory (ROM) during programming and erasure. To prevent destruction of the chip, ensure that the following operations are not performed during programming and erasure.

- Cutting off the power supply
- Transitions to software standby mode
- Read access to the flash memory by the CPU, DMAC or DTC
- Writing a new value to the FRQCR register
- Setting the PCKAR register for a different frequency from that of Pϕ.

27.10.11 Abnormal Ending of Programming or Erasure

A lock bit may be set to 0 (in the protected state) due to a reset, an FCU reset by the FRESET bit in the FRESETR register, a transition to the command-locked state because an error has been detected, or programming or erasure not being completed normally.

If this is the case, issue a block erase command to erase the lock bit while the FPROTR.FPROTCN bit is set to 1. After that, repeat the programming until it is finished.

Section 28 Data Flash (FLD)

This LSI includes 32 Kbytes of flash memory (FLD) for storing data. The FLD has the following features.

28.1 Features

- Flash-memory MATs
  Data MAT: 32 Kbytes (8 Kbytes × 4 blocks)

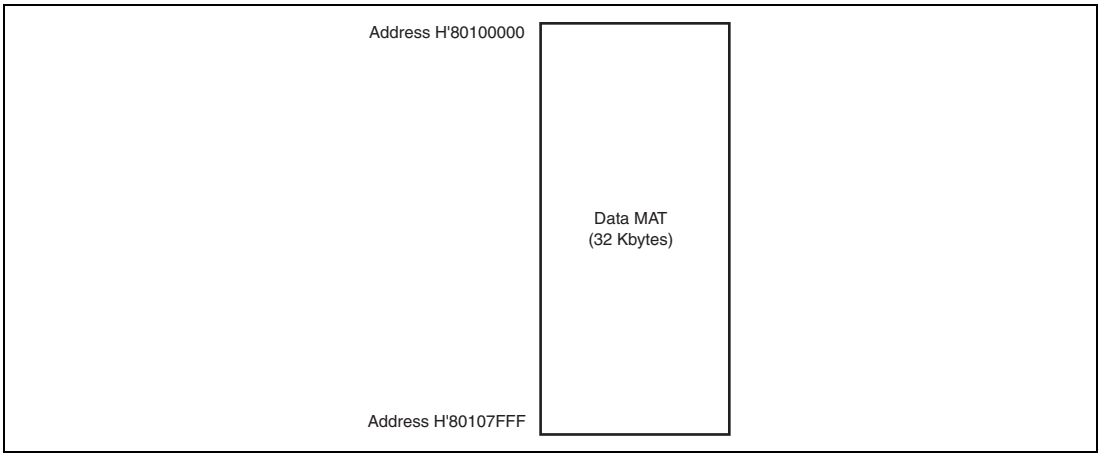


Figure 28.1 Memory MAT Configuration in FLD

- Reading through the peripheral bus (P bus)

  The data MAT can be read through the P bus. Reading programs can be executed on the on-chip RAM or on-chip ROM.

- Programming and erasing methods

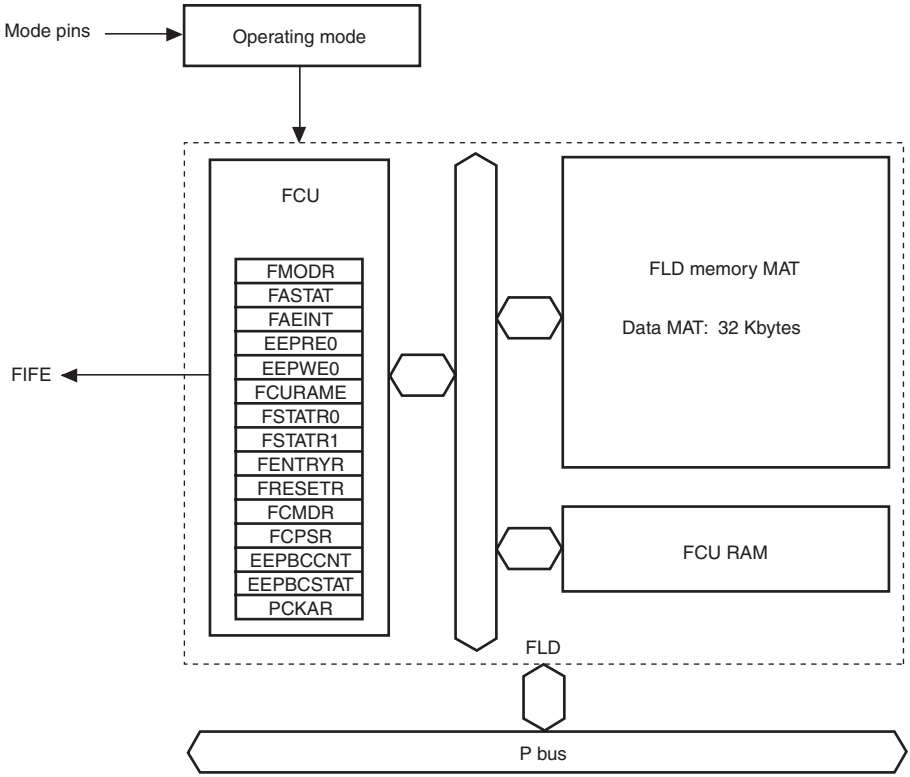
  The FLD has a dedicated sequencer (FCU) for reprogramming of the flash-memory MATs. The ROM is programmed and erased by issuing commands to the FCU.

- BGO (background operation) function

  1. The CPU can execute programs located in areas other than the ROM while the FCU is programming or erasing the ROM.
  2. A program located in ROM can be executed while the FCU is programming or erasing the data flash.

- Suspending and resuming operation

  After the FCU has suspended programming or erasing the ROM, and the CPU has executed the program in the ROM, the FCU can resume programming or erasure of the ROM. These operations are called suspension (suspend processing) and resumption (resume processing).



[Legend]

FMODR:	Flash mode register
FASTAT:	Flash access status register
FAEINT:	Flash access error interrupt enable register
EEPRE0:	FLD read enable register 0
EEPWE0:	FLD program/erase enable register 0
FCURAME:	FCU RAM enable register
FSTATR0, FSTATR0:	Flash status registers 0 and 1
FENTRYR:	Flash P/E mode entry register
FRESETR:	Flash reset register
FCMDR:	FCU command register
FCPSR:	FCU processing switch register
EEPBCCNT:	FLD blank check control register
EEPBCSTAT:	FLD blank check status register
PCKAR:	Peripheral clock notification register
FIFE:	Flash interface error interrupt

Figure 28.2 Block Diagram of FLD

- Programming/erasing unit

  The data MAT is programmed in 8-byte or 128-byte units and erased in block units (8 Kbytes) in user mode, user program mode, and user boot mode. In boot mode, the data MAT is programmed in 256-Kbyte units and erased in block units (8 Kbytes). The product information MAT is read-only memory and cannot be programmed or erased.

  Figure 28.3 shows the block configuration of the data MAT of this LSI. The data MAT is divided into four 8-Kbyte blocks (DB00 to DB03).

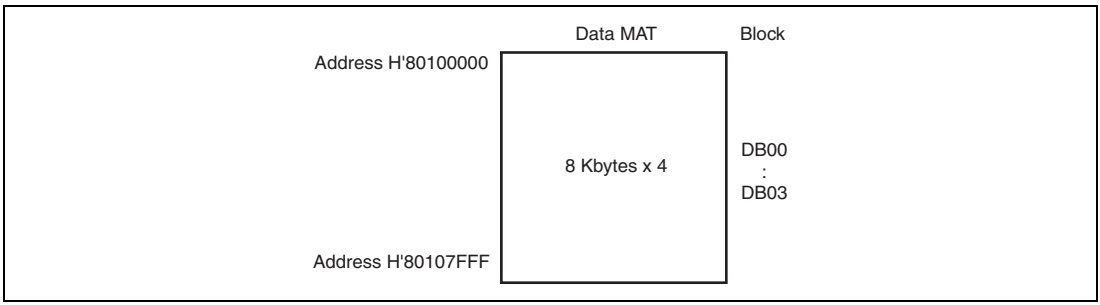


Figure 28.3 Block Configuration of Data MAT

- Blank check function

  If data is read from erased FLD by the CPU, undefined values are read. Using blank check command of the FCU allows checking of whether the FLD is erased (in a blank state). Either an 8 Kbytes (1 erasure block) or 8 bytes of area can be checked by a single execution of the blank check command.

  Blank checking proceeds for areas where erasure has been completed normally to confirm that the data have actually been erased. When erasure or programming in progress is stopped (e.g. by input of the reset signal or shutting down the power), blank checking cannot be used to check whether the data have actually been erased or written.

- Four types of on-board programming modes
  - — Boot mode

    The data MAT can be programmed using the SCI. The bit rate for SCI communications between the host and the LSI can be automatically adjusted.
  - — User mode/user program mode

    The data MAT can be programmed with a desired interface. The user mode includes the MCU extended mode and MCU single-chip mode (modes 2 and 3) in which the on-chip ROM is enabled.

— USB boot mode

A mode for programming via the USB

— User boot mode

The data MAT can be programmed with a desired interface. To make a transition to this mode, a reset is needed.

- Protection modes

  This LSI supports two modes to protect memory against programming, erasing, or reading: hardware protection by the levels on the mode pins and software protection by the setting of the FENTRYD bit, EEPRE0 register, or EEPWE0 register. The FENTRYD bit enables or disables data MAT programming or erasure by the FCU. EEPRE0 controls protection of each data MAT block against reading, and EEPWE0 controls protection against programming and erasure.

  The LSI also provides a function to suspend programming or erasure when abnormal operation is detected during programming or erasure. In addition, the LSI provides a function to protect the FLD against instruction fetch attempted by the CPU.

- Programming and erasing time and count

  Refer to section 33, Electrical Characteristics.

28.2 Input/Output Pins

Table 28.1 shows the input/output pins used for the FLD. The combination of MD1 and MD0 pin levels determines the FLD programming mode (see section 28.4, Overview of FLD-Related Modes). In boot mode, programming and erasing the FLD can be performed by the host via the PA3/RxD1 and PA4/TxD1 pins (refer to section 28.5, Boot Mode).

Table 28.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	This LSI enters the power-on reset state when this signal goes low.
Mode	MD1, MD0	Input	These pins specify the operating mode.
Receive data in SCI channel 1	PA3/RxD1	Input	Receives data through SCI channel 1 (communications with host)
Transmit data in SCI channel 1	PA4/TxD1	Output	Transmits data through SCI channel 1 (communications with host)
Pull-up control	PUPD (PB15)	Output	Pull-up control (used in USB boot mode)
USB data	USD+ USD-	I/O	USD signal from the USB with a transceiver (used in USB boot mode)
USB cable connection monitor	VBUS	Input	Detects connection and disconnection of the USB cable (used in USB boot mode)
USB clock select	PB14	Input	Selects the clock supplied by the USB (used in USB boot mode)

28.3 Register Descriptions

Table 28.2 shows the FLD-related registers. Some of these registers have ROM-related bits, but this section only describes the FLD-related bits. For the registers consisting of bits used by the ROM and FLD in common (FCURAME, FSTATR0, FSTATR1, FRESETR, FCMDR, and FCPSR) and the ROM-dedicated bits, refer to section 27.3, Register Descriptions. The FLD-related registers are initialized by a power-on reset.

Table 28.2 Register Configuration

Register Name	Symbol	R/W*1	Initial Value	Address	Access Size
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*2	H'00	H'FFFFA810	8
Flash access error interrupt enable register	FAEINT	R/W	H'9F	H'FFFFA811	8
FLD read enable register 0	EEPRE0	R/(W)*3	H'0000	H'FFFFA840	8, 16
FLD program/erase enable register 0	EEPWE0	R/(W)*3	H'0000	H'FFFFA850	8, 16
FCU RAM enable register	FCURAME	R/(W)*3	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*5	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*5	H'FFFFA901	16
Flash P/E mode entry register	FENTRYR	R/(W)*4	H'0000*5	H'FFFFA902	8, 16
Flash reset register	FRESETR	R/(W)*3	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*5	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W	H'0000*5	H'FFFFA918	8, 16
FLD blank check control register	EEPBCCNT	R/W	H'0000*5	H'FFFFA91A	8, 16
FLD blank check status register	EEPBCSTAT	R	H'0000*5	H'FFFFA91E	8, 16
Peripheral clock notification register	PCKAR	R/W	H'0000*5	H'FFFFA938	8, 16

Notes: 1. In on-chip ROM disabled mode, the bits of the FLD-related registers are always read as 0 and writing to them is ignored.
2. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
3. This register can be written to only when a specified value is written to the upper byte in word access. The data written to the upper byte is not stored in the register.
4. This register can be written to only when a specified value is written to the upper byte in word access; the register is initialized when a value not allowed for the register is written to the upper byte. The data written to the upper byte is not stored in the register.
5. This register can be initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

28.3.1 Flash Mode Register (FMODR)

FMODR specifies an operating mode for the FCU. In on-chip ROM disabled mode, the FMODR bits are always read as H'00, and writing to them is ignored. FMODR can be initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	FRDMD	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	FRDMD	0	R/W	FCU Read Mode Select Bit Selects the read mode to read the ROM or FLD using FCU. This bit specifies the FLD lock bit read mode transition or blank check processing in the FLD (see section 28.6.1, FCU Command List, 28.6.3, FCU Command Usage), whereas this bit must be set to specify the read method for the lock bits in the ROM (see section 27, Flash Memory (ROM)). 0: Memory area read mode This mode is selected to enter the FLD lock bit read mode. Since the FLD has no lock bits, reading an FLD area results in an undefined value. 1: Register read mode To make the blank check command available for use, register read mode is set.
3 to 0	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.

28.3.2 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and FLD. In on-chip ROM disabled mode, FASTAT is read as H'00 and writing to it is ignored. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 28.7.3, Error Protection). To cancel command-locked state, set FASTAT to H'10, and then issue a status-clear command to the FCU. FASTAT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written to clear the flag after 1 is read.

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAE	0	R/(W)*	ROM Access Error Refer to section 27, Flash Memory (ROM).
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLK	0	R	FCU Command Lock Indicates whether the FCU is in command-locked state (see section 28.7.3, Error Protection). 0: The FCU is not in command-locked state 1: The FCU is in command-locked state [Setting condition] • The FCU detects an error and enters command-locked state. [Clearing condition] • The FCU completes the status-clear command processing.

Bit	Bit Name	Initial Value	R/W	Description
3	EEPAE	0	R/(W)*	FLD Access Error Indicates whether an access error has been generated for the FLD. If this bit becomes 1, the ILGLERR bit in FSTATR0 is set to 1 and the FCU enters command-locked state. 0: No FLD access error has occurred 1: An FLD access error has occurred [Setting conditions] • A read access command is issued to the FLD area while the FENTRYD bit in FENTRYR is 1 in FLD P/E normal mode. • A write access command is issued to the FLD area while the FENTRYD bit in FENTRYR is 0. • An access command is issued to the FLD area while the FENTRY0 bit in FENTRYR is 1. [Clearing condition] • 0 is written to this bit after reading EEPAE = 1.
2	EEPIFE	0	R/(W)*	FLD Instruction Fetch Error Indicates whether an instruction fetch error has been generated for the FLD. 0: No FLD instruction fetch error has occurred 1: An FLD instruction fetch error has occurred [Setting condition] • An attempt is made to fetch an instruction from the FLD. [Clearing condition] • 0 is written to this bit after reading EEPIFE = 1.

Bit	Bit Name	Initial Value	R/W	Description
1	EEPRPE	0	R/(W)*	FLD Read Protect Error Indicates whether an error has been generated against the FLD read protection provided by the EEPRE0 and EEPRE1 settings. 0: The FLD has not been read against the EEPRE0 setting 1: An attempt has been made to read data from the FLD against the EEPRE0 setting [Setting condition] • An attempt is made to read data from the FLD area that has been read-protected through the EEPRE0 setting. [Clearing condition] • 0 is written to this bit after reading EEPRPE = 1.
0	EEPWPE	0	R/(W)*	FLD Program/Erase Protect Error Indicates whether an error has been generated against the FLD program/erasure protection provided by the EEPWE0 setting. 0: No programming or erasing command has been issued to the FLD against the EEPWE0 setting 1: A programming or erasing command has been issued to the FLD against the EEPWE0 setting [Setting condition] • A programming or erasing command is issued to the FLD area that has been program/erase-protected through the EEPWE0 setting. [Clearing condition] • 0 is written to this bit after reading EEPWPE = 1.

Note: * Only 0 can be written to clear the flag after 1 is read.

28.3.3 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupt requests. In on-chip ROM disabled mode, FAEINT is read as H'00 and writing to it is ignored. FAEINT is initialized by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
Initial value:	1	0	0	1	1	1	1	1
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ROMAEIE	1	R/W	ROM Access Error Interrupt Enable Refer to section 27, Flash Memory (ROM).
6, 5	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
4	CMDLKIE	1	R/W	FCU Command Lock Interrupt Enable Enables or disables an FIFE interrupt request when FCU command-locked state is entered and the CMDLK bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when CMDLK = 1 1: Generates an FIFE interrupt request when CMDLK = 1
3	EEPAEIE	1	R/W	FLD Access Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD access error occurs and the EEPAE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPAE = 1 1: Generates an FIFE interrupt request when EEPAE = 1

Bit	Bit Name	Initial Value	R/W	Description
2	EEPIFEIE	1	R/W	FLD Instruction Fetch Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD instruction fetch error occurs and the EEPIFE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPIFE = 1 1: Generates an FIFE interrupt request when EEPIFE = 1
1	EEPRPEIE	1	R/W	FLD Read Protect Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD read protect error occurs and the EEPRPE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPRPE = 1 1: Generates an FIFE interrupt request when EEPRPE = 1
0	EEPWPEIE	1	R/W	FLD Program/Erase Protect Error Interrupt Enable Enables or disables an FIFE interrupt request when an FLD program/erase protect error occurs and the EEPWPE bit in FASTAT becomes 1. 0: Does not generate an FIFE interrupt request when EEPWPE = 1 1: Generates an FIFE interrupt request when EEPWPE = 1

28.3.4 FLD Read Enable Register 0 (EEPRE0)

EEPRE0 enables or disables read access to blocks DB00 to DB03 (see figure 28.3) in the data MAT. In on-chip ROM disabled mode, EEPRE0 is read as H'0000 and writing to it is ignored. EEPRE0 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	DBRE03	DBRE02	DBRE01	DBRE00
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R/W	R/W	R/W	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	All 0	R/(W)*	Key Code These bits enable or disable DBRE03 to DBRE00 bit modification. The data written to these bits are not stored.
7 to 4	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
3	DBRE03	0	R/W	DB03 to DB00 Block Read Enable Enables or disables read access to blocks DB03 to DB00 in the data MAT. The DBREi bit (i = 03 to 00) controls read access to block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'2D is written to the KEY bits. 0: Disables read access 1: Enables read access
2	DBRE02	0	R/W	
1	DBRE01	0	R/W	
0	DBRE00	0	R/W	

Note: * Write data is not retained.

28.3.5 FLD Program/Erase Enable Register 0 (EEPWE0)

EEPWE0 enables or disables programming and erasure of blocks DB00 to DB03 (see figure 28.3) in the data MAT. In on-chip ROM disabled mode, EEPWE0 is read as H'0000 and writing to it is ignored. EEPWE0 is initialized by a power-on reset.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	KEY								—	—	—	—	DBWE03	DBWE02	DBWE01	DBWE00
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R/W	R/W	R/W	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	KEY	All 0	R/(W)*	Key Code These bits enable or disable DBWE03 to DBWE00 bit modification. The data written to these bits are not stored.
7 to 4	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
3	DBWE03	0	R/W	DB03 to DB00 Block Program/Erase Enable Enables or disables programming and erasure of blocks DB03 to DB00 in the data MAT. The DBWEi bit (i = 03 to 00) controls programming and erasure of block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'1E is written to the KEY bits. 0: Disables programming and erasure 1: Enables programming and erasure
2	DBWE02	0	R/W	
1	DBWE01	0	R/W	
0	DBWE00	0	R/W	

Note: * Write data is not retained.

28.3.6 Flash P/E Mode Entry Register (FENTRYR)

FENTRYR specifies the P/E mode for the ROM or FLD. To specify the P/E mode for the ROM or FLD so that the FCU can accept commands, set either FENTRYD or FENTRY0 to 1. In on-chip ROM disabled mode, FENTRYR is read as H'0000 and writing to it is ignored. FENTRYR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

In access to the FENTRYR for a mode transition of the FCU, write to the register and then read it. Proceed with programming, erasure or reading of the FLD after confirming the register setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FEKEY								FENTRYD	—	—	—	—	—	—	FENTRY0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R	R	R	R	R	R	R/W

Note: * Write data is not retained.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	FEKEY	H'00	R/(W)*	Key Code These bits enable or disable the FENTRYD and FENTRY0 bit modification. The data written to these bits are not retained.

Bit	Bit Name	Initial Value	R/W	Description
7	FENTRYD	0	R/W	FLD P/E Mode Entry This bit specifies the P/E mode for the FLD. 00: The FLD is in read mode 11: The FLD is in P/E mode [Write enabling conditions] When the following conditions are all satisfied: • The LSI is in on-chip ROM enabled mode. • The FRDY bit in FSTATR0 is 1. • H'AA is written to FEKEY in word access. [Setting condition] • 1 is written to FENTRYD while the write enabling conditions are satisfied and FENTRYR is H'0000. [Clearing conditions] • This register is written to in byte access. • A value other than H'AA is written to FEKEY in word access. • 0 is written to FENTRYD while the write enabling conditions are satisfied. • FENTRYR is written to while FENTRYR is not H'0000 and the write enabling conditions are satisfied.
6 to 1	—	All 0	R	Reserved The write value should always be 0; otherwise normal operation cannot be guaranteed.
0	FENTRY0	0	R/W	ROM P/E Mode Entry 1, 0 Refer to section 27, Flash Memory (ROM).

Note: * Write data is not retained.

28.3.7 FLD Blank Check Register (EEPBCCNT)

EEPBCCNT specifies the addresses and sizes of the target areas to be checked by the blank check command. In on-chip ROM disabled mode, EEPBCCNT is read as H'0000, and writing to it is ignored. EEPBCCNT is initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	BCADR										-	-	BC SIZE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved The write value must always be 0; otherwise operation is not guaranteed.
12 to 3	BCADR	All 0	R/W	Blank Check Address Setting Bit Use these bits to specify the address of the target area when the size of the target area to be checked by the blank check command is 8 bytes (the BCSIZE bit is set to 0). When the BCSIZE bit is set to 0, the start address of the target area is the value obtained by summing the EEPBCCNT value (the value obtained by shifting the set BCADR value by 3 bits) and the start address of an erased block specified when a blank check command is issued.

Bit	Bit Name	Initial Value	R/W	Description
2, 1	—	All 0	R	Reserved The write value must always be 0; otherwise operation is not guaranteed.
0	BCSIZE	0	R/W	Blank Check Size Setting Bit This bit selects the size of the target area to be checked by the blank check command. 0: Selects 8 bytes as the size of a blank check target area. 1: Selects 8 Kbytes as the size of a blank check target area.

28.3.8 FLD Blank Check Status Register (EEPBCSTAT)

EEPBCSTAT stores check results by executing the blank check command. In on-chip ROM disabled mode, EEPBCSTAT is read as H'0000, and writing to it is ignored. EEPBCSTAT is initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BCST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	—	All 0	R	Reserved The write value must be 0; otherwise operation is not guaranteed.
0	BCST	0	R	Blank Check Status Bit Indicates the result of a blank check. 0: The target area is erased (blank). 1: The target area is filled with 0s and/or 1s.

28.4 Overview of FLD-Related Modes

Figure 28.4 shows the FLD-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD1 and MD0 pin settings, refer to section 3, MCU Operating Modes.

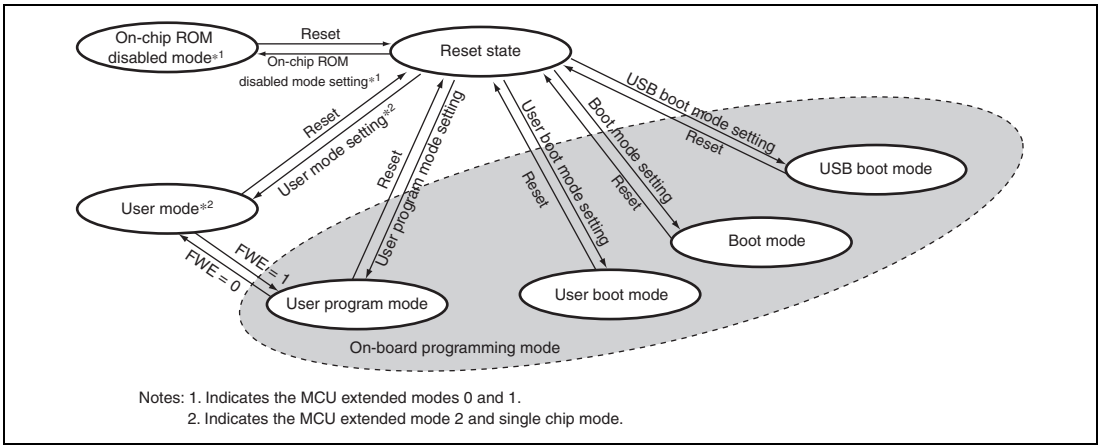


Figure 28.4 FLD-Related Mode Transition

- The FLD cannot be read, programmed, or erased in on-chip ROM disabled mode.
- The data MAT can be read, programmed, and erased on the board in user mode, user program mode, user boot mode, boot mode, and USB boot mode.
- In user mode, the ROM cannot be programmed or erased but the FLD can be programmed and erased. While the FLD is being programmed or erased, the ROM can be read. Therefore, the user can program the FLD while executing an application program in the ROM protected against programming and erasure.

Table 28.3 compares programming- and erasure-related items for the boot mode, user mode, user program mode, user boot mode, and USB boot mode.

Table 28.3 Comparison of Programming Modes

Item	Boot Mode	User Mode	User Program Mode	User Boot Mode	USB Boot Mode
Programming/ erasure environment	On-board programming				
Programming/ erasure enabled MAT	Data MAT	Data MAT	Data MAT	Data MAT	Data MA
Programming/ erasure control	Host	FCU	FCU	FCU	Host
Entire area erasure	Available (automatic)	Available	Available	Available	Available (automatic)
Block erasure	Available*1	Available	Available	Available	Available*1
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM	From any device via RAM	From host via USB
Reset-start MAT	Embedded program stored MAT	User MAT	User MAT	User boot MAT*2	Embedded program stored MAT

Notes: 1. The entire area is erased when the LSI is started. After that, a specified block can be erased.
2. After the LSI is started in the embedded program stored MAT and the boot program provided by Renesas Corp. is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- In boot mode or USB boot mode, the user MAT and user boot MAT in the ROM and the data MAT are all erased immediately after the LSI is started. The data MAT can then be programmed from the host via the SCI. The data MAT can also be read after this entire area erasure.
- In user boot mode, a boot operation with a desired interface can be implemented through mode pin settings different from those in user mode or user program mode.

28.5 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, refer to section 27, Flash Memory (ROM). This section describes only the commands dedicated for the FLD.

28.5.1 Inquiry/Selection Host Commands

Table 28.4 shows the inquiry/selection host commands dedicated to the FLD. The data MAT inquiry and data MAT information inquiry commands are used in the step for inquiry regarding the MAT programming information shown in figure 27.11 in section 27.5.5, Inquiry/Selection Host Command Wait State.

Table 28.4 Inquiry/Selection Host Commands (for FLD only)

Host Command Name	Function
Data MAT inquiry	Inquires regarding the availability of user 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 this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Data MAT Inquiry

In response to a data MAT inquiry command sent from the host, this LSI returns the information concerning the availability of data MATs.

Command	H'2A

Response	H'3A	Size	Availability	SUM

[Legend]

Size (1 byte): Total number of characters in the availability field (fixed at 1)

Availability (1 byte): Availability of data MATs (fixed at H'01)
H'00: No data MAT is available
H'01: 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, this LSI returns the number of data MATs and their addresses.

Command	H'2B

Response			
	H'3B	Size	MAT count
	MAT start address		
	MAT end address		
	MAT start address		
	MAT end address		
	:		
	MAT start address		
	MAT end address		
	SUM		

[Legend]

Size (1 byte): Total number of bytes in the MAT count, MAT start address, and MAT end address fields

MAT count (1 byte): Number of data MATs (consecutive areas are counted as one MAT)

MAT start address (4 bytes): Start address of a data MAT

MAT end address (4 bytes): End address of a data MAT

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 (refer to section 27.5.5, Inquiry/Selection Host Command Wait State).

28.5.2 Programming/Erasing Host Commands

Table 28.5 shows the programming/erasing host commands dedicated to the FLD. FLD-dedicated host commands are provided only for checksum and blank check; the programming, erasing, and reading commands are used in common for the ROM and FLD.

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 configuration 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 selection, and memory read commands, refer to section 27.5.6, Programming/Erasing Host Command Wait State. For the erasure block information inquiry command, refer to section 27.5.5, Inquiry/Selection Host Command Wait State.

Table 28.5 Programming/Erasure Host Commands (for FLD)

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 this LSI and the "response" indicates a response sent from this LSI to the host. The "checksum" is byte-size data calculated so that the sum of all bytes to be sent by this LSI becomes H'00.

(1) Data MAT Checksum

In response to a data MAT checksum command sent from the host, this LSI sums the data MAT data in byte units and returns the result (checksum).

Command	H'61			
Response	H'71	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, this LSI checks whether the data MAT is completely erased. When the data MAT is completely erased, this LSI returns a response (H'06). If the user MAT has an unerased area, this LSI returns an error response (sends H'E2 and H'52 in that order).

Command	H'62	
Response	H'06	
Error response	H'E2	H'52

28.6 User Mode, User Program Mode, and User Boot Mode

28.6.1 FCU Command List

To program or erase the data MAT in user mode, user program mode, or user boot mode, issue FCU commands to the FCU. Table 28.6 is a list of FCU commands for FLD programming and erasure.

Table 28.6 FCU Command List (FLD-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 28.6.2, Conditions for FCU Command Acceptance).
Status read mode transition	Moves to the status read mode (see section 28.6.2, Conditions for FCU Command Acceptance).
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 28.6.2, Conditions for FCU Command Acceptance).
Program	Programs FLD (in 8-byte or 128-byte units).
Block erase	Erases FLD (in block units).
P/E suspend	Suspends programming or erasure.
P/E resume	Resumes programming or erasure.
Status register clear	Clears the IRGERR, ERSERR, and PRGERR bits in FSTATR0 and cancels the command-locked state.
Blank check	Checks if a specified area is erased (blank).
Peripheral clock notification	Specifies the peripheral clock frequency

FCU commands other than the program command and blank check command are also used for ROM programming and erasure. When the blank check command is issued to the ROM, the lock bits in the ROM are read out.

To issue a command to the FCU, access the FLD area through the P bus. Table 28.7 shows the FCU command formats for the program command and blank check command. For the other command formats, refer to section 27.6.1, FCU Command List. When a P-bus access, as shown in table 28.7, is made under specified conditions, the FCU performs processing specified by a selected command. For the conditions for the FCU command acceptance, refer to section 28.6.2, Conditions for FCU Command Acceptance. For details of command usage, refer to section 28.6.3, FCU Command Usage.

When the FRDMD bit is set to 0 (memory area read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU accepts the lock bit read mode transition command. Since the FLD has no lock bits, making P-bus access after a transition to the lock bit read mode results in undefined read data. The FCU detects no access violation error when the undefined data is read. When the FRDMD bit is set to 1 (register read mode), if the data in the first cycle of an FCU command is determined as H'71, the FCU enters a waiting state to wait for the command in the second cycle (H'D0) of the blank check command. At this stage, if H'D0 is written into an FLD area by a P-bus write access, the FCU detects it and starts performing the blank check processes specified by the set values in the EEPBCCNT register, and once the check completes the FCU writes check results into the EEPBCSTAT register.

There are two suspending modes to be initiated by the P/E suspend command; the suspension-priority mode and erasure-priority mode. For details of each mode, refer to section 27.6.4, Suspending Operation.

Table 28.7 FCU Command Formats (for FLD only)

Command	Number of Bus Cycles	First Cycle		Second Cycle		Third Cycle		Fourth Cycle to Cycle N + 2		Cycle N + 3	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Program (8-byte programming: N = 4)	7	EA	H'E8	EA	H'04	WA	WD1	EA	WDn	EA	H'D0
Program (128-byte programming: N = 64)	67	EA	H'E8	EA	H'40	WA	WD1	EA	WDn	EA	H'D0
Blank check	2	EA	H'71	BA	H'D0	—	—	—	—	—	—

[Legend]

EA: FLD area address
An arbitrary address within the range of H'8010000 to H’80107FFF

WA: The start address of write data

BA: The address of an FLD erasure block
(An arbitrary address in the erase target block)

WDn: n-th word of programming data (n = 1 to N)

28.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 28.5 is an FCU mode transition diagram.

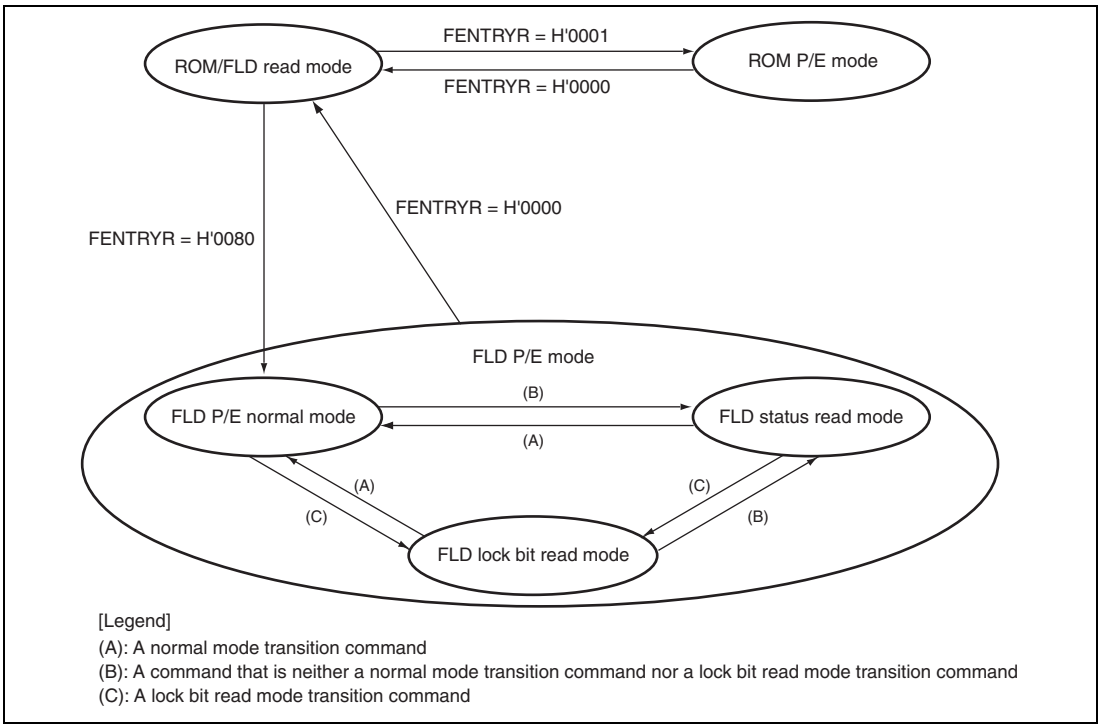


Figure 28.5 FCU Mode Transition Diagram (FLD-Related Modes)

(1) ROM P/E Mode

The FCU can accept ROM programming and erasing commands in this mode. The FLD cannot be read. The FCU enters this mode when the FENTRYD bit is set to 0 and the FENTRY0 bit is set to 1 in FENTRYR. For details of this mode, refer to section 27.6.2, Conditions for FCU Command Acceptance.

(2) ROM/FLD Read Mode

The FLD can be read through the HPB, and the ROM can be read through the ROM cache at a high speed. The FCU does not accept commands. The FCU enters this mode when the FENTRY0 bit is set to 0 and the FENTRYD bit in FENTRYR is set to 0.

(3) FLD P/E Mode

- FLD P/E normal mode

  The FCU enters this mode when the FENTRYD bit is set to 1 and the FENTRY0 bit is set to 0 in ROM/FLD read mode or ROM P/E mode, or when a normal mode transition command is accepted in FLD P/E mode. Table 28.8 shows the commands that can be accepted in this mode. If the FLD area is read through the P bus, an FLD access error occurs and the FCU enters the command-locked state.
- FLD status read mode

  The FCU enters this mode when the FCU accepts a command that is neither the normal mode transition command nor the lock bit read mode transition command in FLD P/E mode. The FLD status read mode includes the state in which the FRDY bit in FSTATR0 is 0 and the command-locked state after an error has occurred. Table 28.8 shows the commands that can be accepted in this mode. If the FLD area is read through the P bus, the FSTATR0 value is read.
- FLD lock bit read mode

  The FCU enters this mode when the FCU accepts a lock bit read mode transition command in FLD P/E mode. Table 28.8 shows the commands that can be accepted in this mode. Since the FLD has no lock bits, reading an FLD area via the P-bus results in an undefined value. However, no access violation occurs in this case. High-speed read operation is available for ROM.

Table 28.8 shows the correlation between each FCU mode/state and its acceptable commands. When an unacceptable command is issued, the FCU enters the command-locked state (see section 28.7.3, Error Protection).

To make sure that the FCU accepts a command, enter the mode in which the FCU can accept the target command, check the FRDY, ILGLERR, ERSERR, and PRGERR bit values in FSTATR0, and the FCUERR bit values in FSTATR1, and then issue the target FCU command. The CMDLK bit in FASTAT holds a value obtained by logical ORing the ILGLERR, ERSERR, and PRGERR bit values in FSTATR0 and the FCUERR bit values in the FSTATR1. Therefore the FCU's error occurrence state can be checked by reading the CMDLK bit. In table 28.8, the CMDLK bit is used as the bit to indicate the error occurrence state. The FRDY bit of FSTATR0 is 0 during the programming/erasure, programming/erasure suspension, and blank check processes. While the FRDY bit is 0, the P/E suspend command can be accepted only when the SUSRDY bit in FSTATR0 is 1.

Table 28.8 includes 0 and 1 in single cells of the ERSSPD, PRGSPD, and FRDY bit rows for the sake of simplification. The ERSSPD bits 1 and 0 indicate the erasure suspension and programming suspension processes, respectively. The PRGSPD bits 1 and 0 indicate the programming suspension and erasure suspension processes, respectively. The FRDY bit value can be either 1 or 0, which is a value held by the bit prior to a transition to the command lock state.

Table 28.8 FCU Modes/States and Acceptable Commands

	P/E Normal Mode			Status Read Mode							Lock Bit Read Mode		
Item	Programming-Suspended	Erasure-Suspended	Other State	Programming/Erasure Processing	Programming/Erasure Suspension Processing	Blank Check Processing	Programming-Suspended	Erasure-Suspended	Command-Locked	Other State	Programming-Suspended	Erasure-Suspended	Other State
FRDY bit in FSTATR0	1	1	1	0	0	0	1	1	0/1	1	1	1	1
SUSRDY bit in FSTATR0	0	0	0	1	0	0	0	0	0	0	0	0	0
ERSSPD bit in FSTATR0	0	1	0	0	0/1	0	0	1	0	0	0	1	0
PRGSPD bit in FSTATR0	1	0	0	0	0/1	0	1	0	0	0	1	0	0
CMDLK bit in FASTAT	0	0	0	0	0	0	0	0	1	0	0	0	0
Normal mode transition	A	A	A	×	×	×	A	A	×	A	A	A	A
Status read mode transition	A	A	A	×	×	×	A	A	×	A	A	A	A
Lock bit read mode transition (lock bit read 1)	A	A	A	×	×	×	A	A	×	A	A	A	A
Program	×	*	A	×	×	×	×	*	×	A	×	*	A
Block erase	×	×	A	×	×	×	×	×	×	A	×	×	A
P/E suspend	×	×	×	A	×	×	×	×	×	×	×	×	×
P/E resume	A	A	×	×	×	×	A	A	×	×	A	A	×
Status register clear	A	A	A	×	×	×	A	A	A	A	A	A	A
Blank check	A	A	A	×	×	×	A	A	×	A	A	A	A
Peripheral clock notification	×	×	A	×	×	×	×	×	×	A	×	×	A

[Legend]

A: Acceptable

*: Only programming is acceptable for the areas other than the erasure-suspended block

×: Not acceptable

28.6.3 FCU Command Usage

This section shows how to program and erase the FLD using the program command and block erase command, respectively, and how to check the erasure status of the FLD using the blank check command. For the firmware transfer to the FCU RAM and the other FCU command usage, refer to section 27.6.3, FCU Command Usage.

If the FCU enters the command lock state in the middle of its handling of commands by setting the FCUERR bit in FSTATR1 to 1, the FRDY bit in FSTATR0 retains 0. Since the FCU halts its operation in the command lock state, the FRDY bit is not set to 1 from 0.

If the FRDY bit retains 0 for longer than the programming/erasure time or suspend delay time (see section 33, Electrical Characteristics), an abnormal operation may have occurred. In such case, initialize the FCU by issuing an FCU reset.

If the FRDY bit is set to 1 upon the termination of an FCU command operation, the FCUERR bit is cleared to 0. On the other hand, it can be checked via the ILGLERR, ERSERR, or PRGERR bit whether or not an error has occurred after a command operation terminates.

(1) Using the Peripheral Clock Notification Command

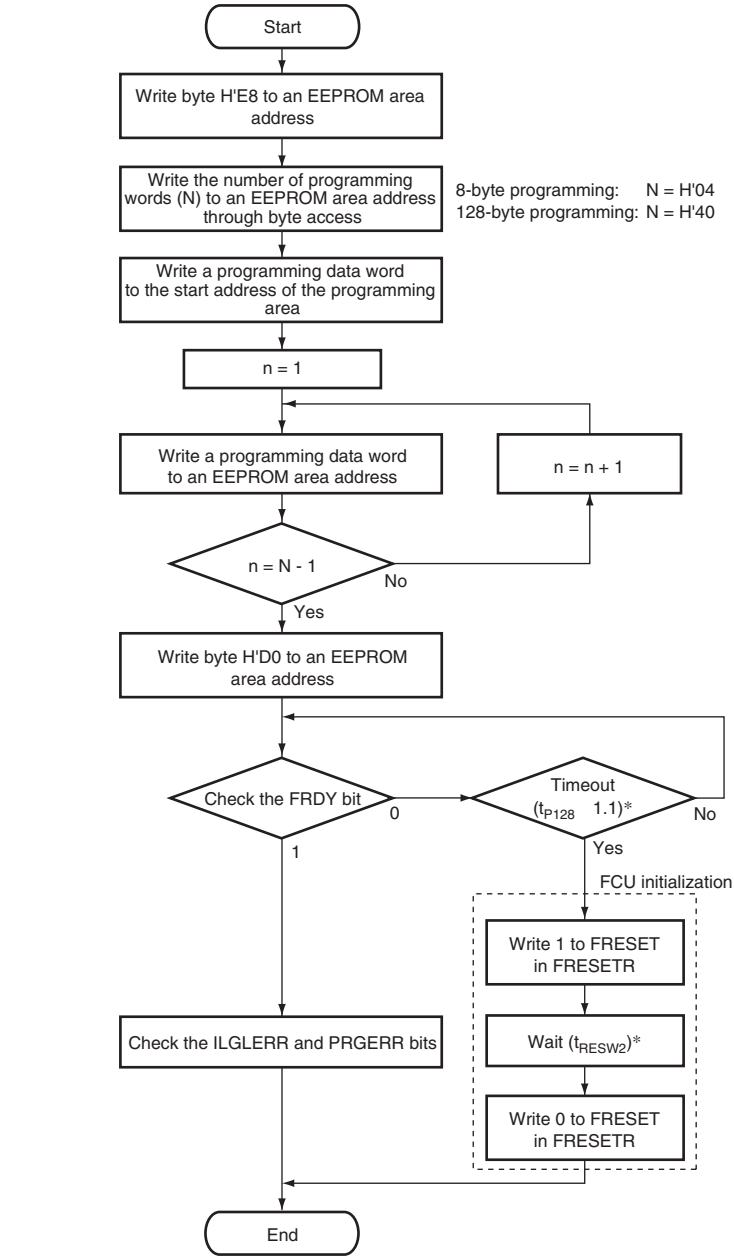
The command is used for notification of the peripheral clock frequency. For details, see section 27.6.3, FCU Command Usage, in section 27, Flash Memory (ROM). Proceed by setting the FENTRYD bit in FENTRYR to 1 and specifying the address as an address within the region corresponding to the data flash (FLD).

(2) Programming

To program the FLD, use the program command. Write byte H'E8 to an FLD area address in the first cycle of the program command and the number of words (N)* to be programmed through byte access in the second cycle. Access the P bus in words from the third cycle to cycle N + 2 of the command. In the third cycle, write the programming data to the start address of the target programming area. Here, the start address must be an 8-byte boundary address for 8-byte programming or a 128-byte boundary address for 128-byte programming. After writing words to FLD area addresses N times, write byte H'D0 to an FLD area address in cycle N + 3; the FCU then starts FLD programming. Read the FRDY bit in FSTATR0 to confirm that FLD programming is completed.

If the area accessed in the third cycle to cycle N + 2 includes addresses that do not need to be programmed, write H'FFFF as the programming data for those addresses. To ignore the programming and erasure protection provided by the EEPWE0 and EEPWE1 settings, set the program/erase enable bit for the target block to 1 before starting programming. To ignore the protection provided by the lock bit during programming, set the FPROTCN bit in FPROTR to 1 before starting programming. Figure 28.6 shows the procedure for FLD programming

Note: * N = H'04 for 8-byte programming or N = H'40 for 128-byte programming.



Notes: * t_{P128}: Time required for programming 128-byte data (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 28.6 Procedure for FLD Programming

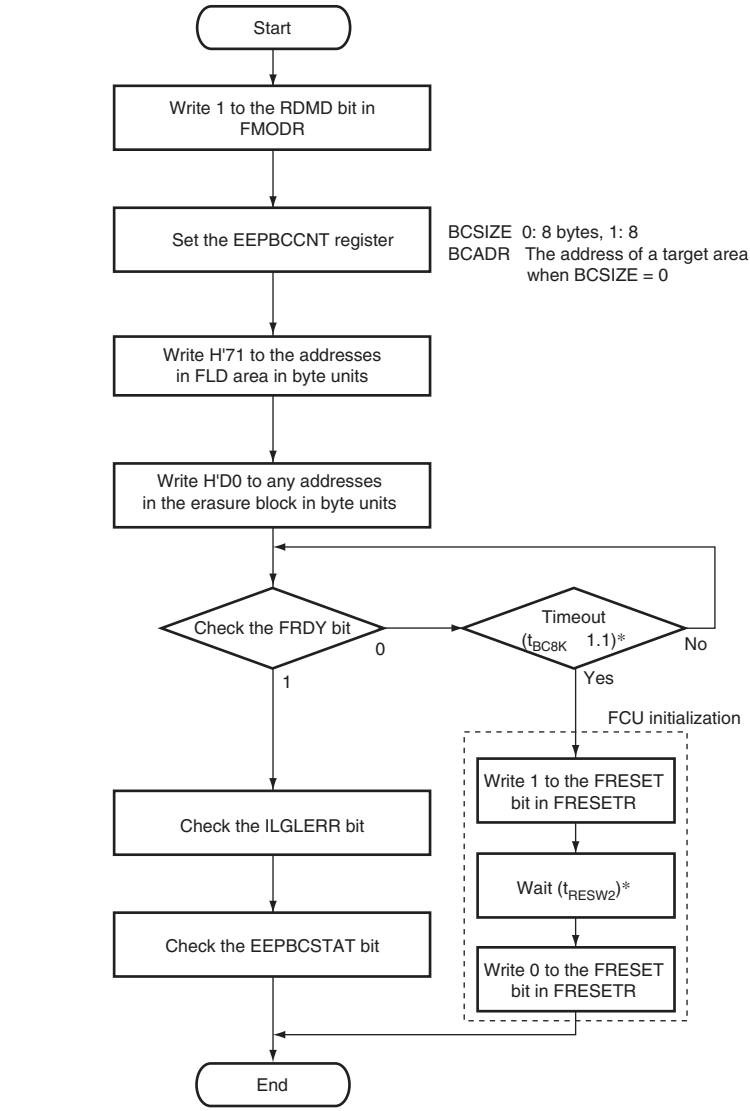
(3) Erasure

To erase the ROM, use the block erase command. The FLD can be erased in the same way as ROM erasure (refer to section 27, Flash Memory (ROM)). Note that the FLD has a programming and erasure protection function through a register. To ignore the programming and erasure protection provided by the EEPWE0 setting, set the program/erase enable bit for the target block to 1 before starting erasure.

(4) Checking of the Erased State

Since reading the FLD erased by the CPU results in undefined values, the blank check command should be used to check the erased state of the FLD. To make the blank check command available for use, set the FRDMD bit in FMODR to 1 to enable the command first, and then specify the size and start address of a target area via the EEPBCCNT register. When the BCSIZE bit of the EEPBCCNT register is set to 1, a check can be performed on the entire erased block (8 Kbytes) specified in the second cycle of the command. When the BCSIZE bit is set to 0, a check can be performed on an 8-byte area starting from the address obtained by summing the start address of the erased area specified in the second cycle of the command and the value held by the EEPBCCNT register. In the first cycle of the command, a value of H'71 is written in byte into an address of the FLD. In the second cycle, once a value of H'D0 is written into a specified address included in the target area, the FCU starts the blank check on the FLD. It can be checked whether or not the check is complete via the FRDY bit in the FSTATR0. After the blank check is complete, it can be checked whether the target area is erased or filled with 0s and/or 1s via the BCST bit of the EEPBCSTAT register.

Figure 28.7 shows the procedure of the FLD blank check.



Notes: * t_{BC8K}: Time required for blank check (see section 33, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 33, Electrical Characteristics).

Figure 28.7 Procedure of the FLD Blank Check

28.7 Protection

There are three types of FLD programming/erasure protection: hardware, software, and error protection.

28.7.1 Hardware Protection

The hardware protection function disables FLD programming and erasure according to the mode pin settings in this LSI.

While the on-chip ROM is disabled, FLD programming, erasing, and reading are disabled. For the operating modes set through the mode pins of this LSI, refer to section 3, MCU Operating Modes.

28.7.2 Software Protection

The software protection function disables FLD programming and erasure according to the control register settings. If an attempt is made to issue a programming or erasing command to the FLD against software protection, the FCU detects an error and enters command-locked state.

(1) Protection through FENTRYR

When the FENTRYD bit in FENTRYR is 0, the FCU does not accept commands for the FLD, so FLD programming and erasure are disabled. If an attempt is made to issue an FCU command for the FLD while the FENTRYD bit is 0, the FCU detects an illegal command error and enters command-locked state (see section 28.7.3, Error Protection).

(2) Protection through EEPWE0

When the DBWEi (i = 00 to 03) bit in EEPWE0 is 0, programming and erasure of block DBi in the data MAT is disabled. If an attempt is made to program or erasure block DBi while the DBWEi bit is 0, the FCU detects a program/erase protect error and enters command-locked state (see section 28.7.3, Error Protection).

28.7.3 Error Protection

The error protection function detects an illegal FCU command issued, an illegal access, or an FCU malfunction, and disables FCU command acceptance (command-locked state). While the FCU is in command-locked state, the FLD cannot be programmed or erased. To cancel command-locked state, issue a status register clear command while FASTAT is H'10.

While the CMDLKIE bit in FAEINT is 1, a flash interface error (FIFE) interrupt is generated if the FCU enters command-locked state (the CMDLK bit in FASTAT becomes 1). While an FLD-related interrupt enable bit (EEPAEIE, EEPIFEIE, EEPRPEIE, or EEPWPEIE) in FAEINT is 1, an FIFE interrupt is generated if the corresponding status bit (EEPAE, EEPIFE, EEPRPE, or EEPWPE) in FASTAT becomes 1.

Table 28.9 shows the error protection types for the FLD and the status bit values (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the EEPAE, EEPIFE, EEPRPE, and EEPWPE bits in FASTST) after each error detection. For the error protection types used in common by the ROM and FLD (FENTRYR setting error, most of illegal command errors, erasing error, programming error, and FCU error), refer to section 27.9.3, Error Protection. If the FCU enters command-locked state due to a command other than a suspend command issued during programming or erasure processing, the FCU continues programming or erasing the FLD. In this state, the P/E suspend command cannot suspend programming or erasure. If a command is issued in command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set due to the previous error detection.

Table 28.9 Error Protection Types (for FLD only

Error	Description	ILGLERR	ERSERR	PRGERR	EEPAE	EEPIFE	EEPRPE	EEPWPE
Illegal command error	The value specified in the second cycle of a program command is neither H'04 nor H'40.	1	0	0	0	0	0	0
	A lock bit program command has been issued to an area in the FLD while the FENTRYD bit of FENTRYR register is set to 1.	1	0	0	0	0	0	0
FLD access error	A read access command has been issued to the FLD area while FENTRYD = 1 in FENTRYR in FLD P/E normal mode.	1	0	0	1	0	0	0
	A write access command has been issued to the FLD area while FENTRYD = 0.	1	0	0	1	0	0	0
	An access command has been issued to the FLD area while the FENTRY0 bit in FENTRYR is 1.	1	0	0	1	0	0	0
FLD instruction fetch error	An instruction fetch has been made in the FLD area.	1	0	0	0	1	0	0
FLD read protect error	A read access command has been issued to the FLD area protected against reading through EEPRE0.	1	0	0	0	0	1	0
FLD program protect error	A program command or block erase command has been issued to the FLD area protected against programming and erasure through EEPWE0.	1	0	0	0	0	0	1

28.8 Usage Notes

28.8.1 Protection of Data MAT Immediately after a Reset

As the initial value of EEPRE0 and EEPWE0 is H'0000, data MAT programming, erasure, and reading are disabled immediately after a reset. To read data from the data MAT, set EEPRE0 appropriately before accessing the data MAT. To program or erase the data MAT, set EEPWE0 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 an error and enters command-locked state.

28.8.2 State in which Interrupts are Ignored

In the following modes or period, the NMI or maskable interrupt requests are ignored.

- Boot mode or USB boot mode
- Programmer mode
- The program in the embedded program stored MAT is being executed immediately after the LSI is started in user boot mode

28.8.3 Programming-/Erasure-Suspended Area

The data stored in the programming-suspended or erasure-suspended area is undetermined. To avoid malfunction due to undefined read data, ensure that no data is read from the programming-suspended or erasure-suspended area.

28.8.4 Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcontrollers

The flash memory programming/erasing program used for conventional F-ZTAT SH microcontrollers does not work with this LSI.

28.8.5 Reset during Programming or Erasure

To reset the FCU by setting the FRESET bit in the FRESETR register during programming or erasure, hold the FCU in the reset state for a period of t_{RESW2} (see section 33, Electrical Characteristics). Since a high voltage is applied to the FLD during programming and erasure, the FCU has to be held in the reset state long enough to ensure that the voltage applied to the memory unit has dropped. Do not read from the FLD while the FCU is in the reset state.

When a power-on reset is generated by asserting the $\overline{RES}$ pin during programming or erasure of the flash memory, hold the reset state for a period of t_{RESW2} (see section 33, Electrical Characteristics). In a power-on reset, not only does the voltage applied to the memory unit have to drop, but the power supply for the FLD and its internal circuitry also have to be initialized. Thus, the reset state must be maintained over a longer period than in the case of resetting the FCU.

When executing a power-on reset by asserting the $\overline{RES}$ pin or the FCU reset with the FRESET bit set in FRESETR during programming/erasure, all data including a lock bit of a programming/erasure target area are undefined.

While programming or erasure is performed, do not generate an internal reset caused by WDT counter overflow. A reset caused by WDT cannot ensure a sufficient time required for voltage drop for the memory unit, initialization of the power supply for the FLD, or initialization of its internal circuit.

28.8.6 Suspension by Programming/Erasure Suspension

When suspending programming/erasure processing with the programming/erasure suspend command, make sure to complete the operations with the resume command.

28.8.7 Prohibition of Additional Programming

One area cannot be programmed twice in succession. To program an area that has already been programmed, be sure to erase the area before reprogramming.

28.8.8 Program for Reading

Execute program code for reading the FLD from on-chip RAM or on-chip ROM.

28.8.9 Items Prohibited during Programming and Erasure

High voltages are applied within the data memory (ROM) during programming and erasure. To prevent destruction of the chip, ensure that the following operations are not performed during programming and erasure.

- Cutting off the power supply
- Transitions to software standby mode
- Read access to the flash memory by the CPU, DMAC or DTC
- Writing a new value to the FRQCR register
- Setting the PCKAR register for a different frequency from that of Pϕ.

28.8.10 Abnormal Ending of Programming or Erasure

A lock bit may be set to 0 (in the protected state) due to a reset, an FCU reset by the FRESET bit in the FRESETR register, a transition to the command-locked state because an error has been detected, or programming or erasure not being completed normally.

If this is the case, issue a block erase command to erase the lock bit while the FPROTR.FPROTCN bit is set to 1. After that, repeat the programming until it is finished.

28.8.11 Handling when Erasure or Programming is Stopped

Checking of areas in which the data have become undefined due to the erasure or programming in progress being stopped (e.g. by input of the reset signal or shutting down the power) to see whether the data have actually been erased or written is not possible. When the data in an area have become undefined, erase the area completely before using it again.

Section 29 On-Chip RAM

The SH7214 and SH7216 Groups incorporate 128-Kbyte RAM, which is connected to F (Fetch), M (Memory), and I (Internal) buses. This on-chip RAM can be accessed via any of these buses independently.

Figure 29.1 shows RAM block diagrams and figure 29.2 shows RAM and bus connections.

The on-chip RAM is allocated in addresses H'FFF80000 to H'FFF9FFFF (pages 0 to 7), as shown in table 29.1.

29.1 Features

- Access

  The CPU/FPU, DMAC, and DTC can access on-chip RAM in 8, 16, or 32 bits. Data in the on-chip RAM can be effectively used as program area or stack area data necessary for access at high speed.

  Four pages (pages 0 to 3): one cycle in case of writing and reading

  Four pages (pages 4 to 7): two cycles in case of writing, three cycles in case of reading
- Ports

  Each page in the on-chip RAM has two independent read and write ports. The read port is connected to I, F, and M buses and the write port is connected to I and M buses. The F and M buses are used for accesses from the CPU. The I bus is used for accesses from external address spaces.
- Priority

  If the same page is accessed from multiple buses simultaneously, the access is performed according to the bus priority. The bus priority is as follows: I bus (highest), M bus (middle), F bus (lowest).
- Pages

  SH72167, SH72147: 128 Kbytes, eight pages (0 to 7 pages)

  SH72166, SH72146: 96 Kbytes, six pages (0 to 5 pages)

  SH72165, SH72145: 64 Kbytes, four pages (0 to 3 pages)

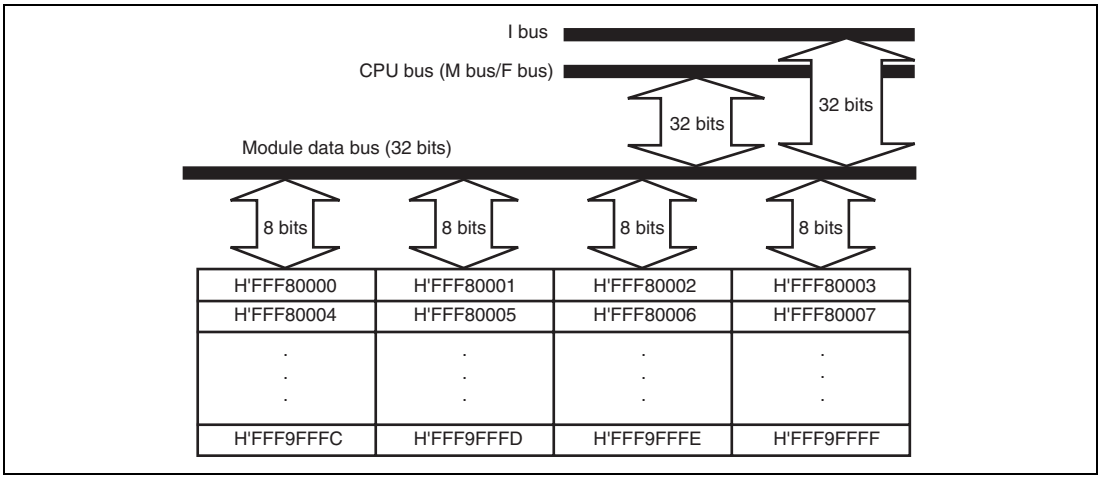


Figure 29.1 RAM Block Diagram

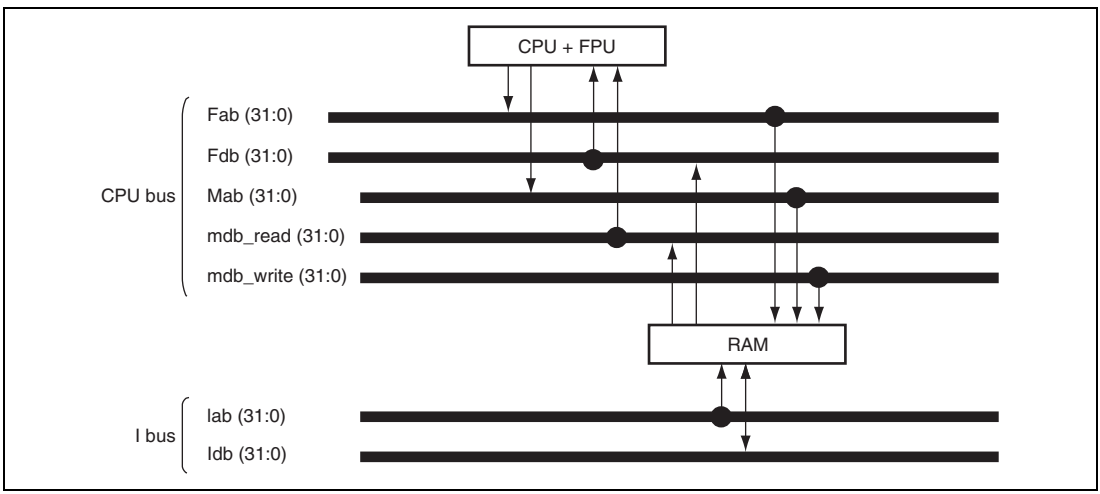


Figure 29.2 Bus Connections in RAM

Table 29.1 On-chip RAM Address Space

Page	Address
Page 0	H'FFF80000 to H'FFF83FFF
Page 1	H'FFF84000 to H'FFF87FFF
Page 2	H'FFF88000 to H'FFF8BFFF
Page 3	H'FFF8C000 to H'FFF8FFFF
Page 4	H'FFF90000 to H'FFF93FFF
Page 5	H'FFF94000 to H'FFF97FFF
Page 6	H'FFF98000 to H'FFF9BFFF
Page 7	H'FFF9C000 to H'FFF9FFFF

29.2 Register Descriptions

The on-chip RAM has registers shown in table 29.2.

Table 29.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
System control register 1	SYSCR1	R/W	H'FF	H'FFFE0402	8
System control register 2	SYSCR2	R/W	H'FF	H'FFFE0404	8

29.2.1 System Control Register 1 (SYSCR1)

SYSCR1 is an 8-bit readable/writable register that enables or disables access to the on-chip RAM. SYSCR1 is initialized to H'FF by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

When an RAME bit is set to 1, the corresponding on-chip RAM area is enabled. When an RAME bit is cleared to 0, the corresponding on-chip RAM area cannot be accessed. In this case, an undefined value is returned when reading data or fetching an instruction from the on-chip RAM, and writing to the on-chip RAM is ignored. The initial value of an RAME bit is 1.

Note that when clearing the RAME bit to 0 to disable the on-chip RAM, be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAME bit. If such an instruction is not executed, the data last written to each page may not be written to the on-chip RAM. Furthermore, an instruction to access the on-chip RAM should not be located immediately after the instruction to write to SYSCR1. If an on-chip RAM access instruction is set, normal access is not guaranteed.

Additionally, note that when setting the RAME bit to 1 to enable the on-chip RAM, be sure to locate an instruction to read SYSCR1 immediately after the instruction to write to SYSCR1. If an on-chip RAM access instruction is set, normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	RAME7	RAME6	RAME5	RAME4	RAME3	RAME2	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
7	RAME7	1	R/W	RAM Enable 7 (corresponding RAM addresses: H'FFF9C000 to H'FFF9FFFF) 0: On-chip RAM disabled 1: On-chip RAM enabled

Bit	Bit Name	Initial Value	R/W	Descriptions
6	RAME6	1	R/W	RAM Enable 6 (corresponding RAM addresses: H'FFF98000 to H'FFF9BFFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
5	RAME5	1	R/W	RAM Enable 5 (corresponding RAM addresses: H'FFF94000 to H'FFF97FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
4	RANME4	1	R/W	RAM Enable 4 (corresponding RAM addresses: H'FFF90000 to H'FFF93FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
3	RAME3	1	R/W	RAM Enable 3 (corresponding RAM addresses: H'FFF8C000 to H'FFF8FFFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
2	RAME2	1	R/W	RAM Enable 2 (corresponding RAM addresses: H'FFF88000 to H'FFF8BFFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
1	RAME2	1	R/W	RAM Enable 1 (corresponding RAM addresses: H'FFF84000 to H'FFF87FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
0	RAME0	1	R/W	RAM Enable 0 (corresponding RAM addresses: H'FFF80000 to H'FFF83FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled

29.2.2 System Control Register 2 (SYSCR2)

SYSCR2 is an 8-bit readable/writable register that enables or disables write to the on-chip RAM. SYSCR2 is initialized to H'FF by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

When an RAMWE bit is set to 1, the corresponding on-chip RAM area is enabled. When an RAMWE bit is cleared to 0, the corresponding on-chip RAM area cannot be written to. In this case, writing to the on-chip RAM is ignored. The initial value of an RAMWE bit is 1.

Note that when clearing the RAME bit to 0 to disable the on-chip RAM, be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAMWE bit. If such an instruction is not executed, the data last written to each page may not be written to the on-chip RAM. Furthermore, an instruction to access the on-chip RAM should not be located immediately after the instruction to write to SYSCR2. If an on-chip RAM access instruction is set, normal access is not guaranteed.

Additionally, note that when setting the RAME bit to 1 to enable the on-chip RAM, be sure to locate an instruction to read SYSCR2 immediately after the instruction to write to SYSCR2. If an on-chip RAM access instruction is set, normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	RAMWE7	RAMWE6	RAMWE5	RAMWE4	RAMWE3	RAMWE2	RAMWE1	RAMWE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
7	RAMWE7	1	R/W	RAM Write Enable 7 (corresponding RAM addresses: H'FFF9C000 to H'FFF9FFFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
6	RAMWE6	1	R/W	RAM Write Enable 6 (corresponding RAM addresses: H'FFF98000 to H'FFF9BFFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled

Bit	Bit Name	Initial Value	R/W	Descriptions
5	RAMWE5	1	R/W	RAM Write Enable 5 (corresponding RAM addresses: H'FFF94000 to H'FFF97FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
4	RAMWE4	1	R/W	RAM Write Enable 4 (corresponding RAM addresses: H'FFF90000 to H'FFF93FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
3	RAMWE3	1	R/W	RAM Write Enable 3 (corresponding RAM addresses: H'FFF8C000 to H'FFF8FFFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
2	RAMWE2	1	R/W	RAM Write Enable 2 (corresponding RAM addresses: H'FFF88000 to H'FFF8BFFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
1	RAMWE1	1	R/W	RAM Write Enable 1 (corresponding RAM addresses: H'FFF84000 to H'FFF87FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
0	RAMWE 0	1	R/W	RAM Write Enable 0 (corresponding RAM addresses: H'FFF80000 to H'FFF83FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled

29.3 Notes on Usage

29.3.1 Page Conflict

If the same page is accessed by the different buses simultaneously, a page conflict occurs. Each of those accesses is handled in such priority scheme as: I bus (highest), M bus (middle), F bus (lowest).

In this case, each access is completed normally but this conflict degrades the memory access efficiency. To avoid this conflict, it is recommended to take preventative measures by software. For example, accessing different memory or different pages using different buses can avoid page conflict.

Section 30 Power-Down Modes

In power-down modes, operation of some of the internal peripheral modules and of the CPU stops. This leads to reduced power consumption. These modes are canceled by a reset or interrupt.

30.1 Features

30.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

1. Sleep mode
2. Software standby mode
3. Module standby function

Table 30.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module states in each mode and the procedures for canceling each mode.

Table 30.1 States of Power-Down Modes

		State*						
Power-Down Mode	**Transition Conditions**	**CPG**	**CPU**	**CPU Register**	**On-Chip Memory**	**On-Chip Peripheral Modules**	**External Memory**	**Canceling Procedure**
Sleep mode	Execute SLEEP instruction with STBY bit cleared to 0 in STBCR	Runs	Halts	Held	Runs	Runs	Auto-refreshing	• Interrupt • Manual reset • Power-on reset • DMA address error
Software standby mode	Execute SLEEP instruction with STBY bit set to 1 in STBCR	Halts	Halts	Held	Halts (contents are held)	Halts	Self-refreshing	• NMI interrupt • IRQ interrupt • Manual reset • Power-on reset
Module standby function	Set the MSTP bits in STBCR2, STBCR3, STBCR4, STBCR5, and STBCR6 to 1	Runs	Runs	Held	Specified module halts (contents are held)	Specified module halts	Auto-refreshing	• Clear MSTP bit to 0 • Power-on reset (only for H-UDI, UBC, and DMAC)

Note: * The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

30.1.2 Reset

A reset is used when the power is turned on or to run the LSI again from the initialized state. There are two types of reset: power-on reset and manual reset. In a power-on reset, all the ongoing processing is halted and any unprocessed events are canceled, and the reset processing starts immediately. On the other hand, a manual reset does not interrupt processing to retain external memory data. Conditions for generating a power-on reset or manual reset are as follows:

(1) Power-On Reset

1. A low level is input to the $\overline{\text{RES}}$ pin.
2. The watchdog timer (WDT) starts counting with the WT/$\overline{\text{IT}}$ bit in WTCSR set to 1 and with the RSTS bit in WRCSR set to 0 while the RSRE bit in WRCSR is 1, and the counter overflows.
3. The H-UDI reset is generated (for details on the H-UDI reset, see section 31, User Debugging Interface (H-UDI)).

(2) Manual Reset

1. A low level is input to the $\overline{\text{MRES}}$ pin.
2. The WDT starts counting with the WT/$\overline{\text{IT}}$ bit in WTCSR set to 1 and with the RSTS bit in WRCSR set to 1 while the RSRE bit in WRCSR is 1, and the counter overflows.

30.2 Input/Output Pins

Table 30.2 lists the pins used for power-down modes.

Table 30.2 Pin Configuration

Name	Pin Name	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	Power-on reset processing starts when a low level is input to this pin.
Manual reset	$\overline{\text{MRES}}$	Input	Manual reset processing starts when a low level is input to this pin.

30.3 Register Descriptions

The following registers are used in power-down modes.

Table 30.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Standby control register	STBCR	R/W	H'00	H'FFFE0014	8
Standby control register 2	STBCR2	R/W	H'00	H'FFFE0018	8
Standby control register 3	STBCR3	R/W	H'7E	H'FFFE0408	8
Standby control register 4	STBCR4	R/W	H'F7	H'FFFE040C	8
Standby control register 5	STBCR5	R/W	H'FF	H'FFFE0418	8
Standby control register 6	STBCR6	R/W	H'DF	H'FFFE041C	8

30.3.1 Standby Control Register (STBCR)

STBCR is an 8-bit readable/writable register that specifies the state of the power-down mode. This register is initialized to H'00 by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is possible.

Bit:	7	6	5	4	3	2	1	0
	STBY	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby Specifies transition to software standby mode. 0: Executing SLEEP instruction puts chip into sleep mode. 1: Executing SLEEP instruction puts chip into software standby mode.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

30.3.2 Standby Control Register 2 (STBCR2)

STBCR2 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR2 is initialized to H'00 by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is possible.

Bit:	7	6	5	4	3	2	1	0
	MSTP10	MSTP9	MSTP8	-	-	-	MSTP4	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP10	0	R/W	Module Stop 10 When the MSTP10 bit is set to 1, the supply of the clock to the H-UDI is halted. 0: H-UDI runs. 1: Clock supply to H-UDI halted.
6	MSTP9	0	R/W	Module Stop 9 When the MSTP9 bit is set to 1, the supply of the clock to the UBC is halted. 0: UBC runs. 1: Clock supply to UBC halted.
5	MSTP8	0	R/W	Module Stop 8 When the MSTP8 bit is set to 1, the supply of the clock to the DMAC is halted. 0: DMAC runs. 1: Clock supply to DMAC halted.
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MSTP4	0	R/W	Module Stop 4 When the MSTP4 bit is set to 1, the supply of the clock to the DTC is halted. 0: DTC runs. 1: Clock supply to DTC halted.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

30.3.3 Standby Control Register 3 (STBCR3)

STBCR3 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR3 is initialized to H'7E by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is possible.

Bit:	7	6	5	4	3	2	1	0
	HIZ	MSTP 36	MSTP 35	-	MSTP 33	MSTP 32	-	MSTP 30
Initial value:	0	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	HIZ	0	R/W	Port High Impedance Selects whether the state of a specified pin is retained or the pin is placed in the high-impedance state in software standby mode. See appendix A, Pin States, to determine the pin to which this control is applied. Do not set this bit when the TME bit of WTSCR of the WDT is 1. When setting the output pin to the high-impedance state, set the HIZ bit with the TME bit being 0. 0: The pin state is held in software standby mode. 1: The pin state is set to the high-impedance state in software standby mode.
6	MSTP36	1	R/W	Module Stop 36 When the MSTP36 bit is set to 1, the supply of the clock to the MTU2S is halted. 0: MTU2S runs. 1: Clock supply to MTU2S halted.
5	MSTP35	1	R/W	Module Stop 35 When the MSTP35 bit is set to 1, the supply of the clock to the MTU2 is halted. 0: MTU2 runs. 1: Clock supply to MTU2 halted.

Bit	Bit Name	Initial Value	R/W	Description
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the supply of the clock to the IIC3 is halted. 0: IIC3 runs. 1: Clock supply to IIC3 halted.
2	MSTP32	1	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the supply of the clock to the ADC0 is halted. 0: ADC0 runs. 1: Clock supply to ADC0 halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP30	0	R/W	Module Stop 30 When the MSTP30 bit is set to 1, the supply of the clock to the flash memory is halted. 0: The flash memory runs. 1: Clock supply to the flash memory halted.

30.3.4 Standby Control Register 4 (STBCR4)

STBCR4 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR4 is initialized to H'F7 by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is possible.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	MSTP44	-	MSTP42	-	MSTP40
Initial value:	1	1	1	1	0	1	1	1
R/W:	R	R	R	R/W	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
4	MSTP44	1	R/W	Module Stop 44 When the MSTP44 bit is set to 1, the supply of the clock to the SCIF3 is halted. 0: SCIF3 runs. 1: Clock supply to SCIF3 halted.
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	MSTP42	1	R/W	Module Stop 42 When the MSTP42 bit is set to 1, the supply of the clock to the CMT is halted. 0: CMT runs. 1: Clock supply to CMT halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP40	1	R	Module Stop 40 When the MSTP40 bit is set to 1, the supply of the clock to the E-DMAC and EtherC is halted. 0: the E-DMAC and EtherC runs. 1: Clock supply to the E-DMAC and EtherC halted.

30.3.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR5 is initialized to H'FF by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is possible.

Bit:	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	MSTP55	-	MSTP53	MSTP52	-	MSTP50
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP57	1	R/W	Module Stop 57 When the MSTP57 bit is set to 1, the supply of the clock to the SCI0 is halted. 0: SCI0 runs. 1: Clock supply to SCI0 halted.
6	MSTP56	1	R/W	Module Stop 56 When the MSTP56 bit is set to 1, the supply of the clock to the SCI1 is halted. 0: SCI1 runs. 1: Clock supply to SCI1 halted.
5	MSTP55	1	R/W	Module Stop 55 When the MSTP55 bit is set to 1, the supply of the clock to the SCI2 is halted. 0: SCI2 runs. 1: Clock supply to SCI2 halted.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
3	MSTP53	1	R/W	Module Stop 53 When the MSTP53 bit is set to 1, the supply of the clock to the SCI4 is halted. 0: SCI4 runs. 1: Clock supply to SCI4 halted.
2	MSTP52	1	R/W	Module Stop 52 When the MSTP52 bit is set to 1, the supply of the clock to the ADC1 is halted. 0: ADC1 runs. 1: Clock supply to ADC1 halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP50	1	R/W	Module Stop 50 When the MSTP50 bit is set to 1, the supply of the clock to the RSPI is halted. 0: the RSPI runs. 1: Clock supply to the RSPI halted.

30.3.6 Standby Control Register 6 (STBCR6)

STBCR6 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR6 is initialized to H'DF by a power-on reset but retains its previous value by a manual reset or in software standby mode. Only byte access is possible.

Bit:	7	6	5	4	3	2	1	0
	USB SEL*1	MSTP 66*2	USB CLK	MSTP 64	-	-	-	-
Initial value:	1	1	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
7	USBSEL*[1]	1	R/W	USB Clock Select Selects the on-chip CPG or the USB oscillator as the source of the USB clock. 0: On-chip CPG 1: USB oscillator
6	MSTP66*[2]	1	R/W	Module Stop 66 When the MSTP66 bit is set to 1, the supply of the clock to the USB is halted. 0: USB runs. 1: Clock supply to USB halted.
5	USBCLK	0	R/W	USB Oscillator Stop When the USBCLK bit is set to 1, the oscillator dedicated for the USB stops. 0: USB oscillator operates. 1: USB oscillator stops.
4	MSTP64	1	R/W	Module Stop 64 When the MSTP64 bit is set to 1, the supply of the clock to the RCAN-ET is halted. 0: RCAN-ET runs. 1: Clock supply to RCAN-ET halted.
3 to 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

Notes: When using the USB, Follow the notes shown below. Otherwise the clock will not be generated correctly so that USB can be operated improperly.

1. When selecting the on-chip CPG, set the frequency of the input clock to 12MHz.
2. When using the USB, set the frequency of the peripheral clock (Pϕ) to 13 MHz or more.

30.4 Operation

30.4.1 Sleep Mode

(1) Transition to Sleep Mode

Executing the SLEEP instruction when the STBY bit in STBCR is 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip modules continue to run in sleep mode. Clock pulses are output continuously on the CK pin.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, IRQ, and on-chip peripheral module), DMA address error, or reset (manual reset or power-on reset).

- Canceling with an interrupt
  When an NMI, IRQ, or on-chip peripheral module interrupt occurs, sleep mode is canceled and interrupt exception handling is executed. When the priority level of the generated interrupt is equal to or lower than the interrupt mask level that is set in the status register (SR) of the CPU, or the interrupt by the on-chip peripheral module is disabled on the module side, the interrupt request is not accepted and sleep mode is not canceled.
- Canceling with a DMAC or DTC address error
  When a DMAC or DTC address error occurs, sleep mode is canceled and DMAC or DTC address error exception handling is executed.
- Canceling with a reset
  Sleep mode is canceled by a power-on reset or a manual reset.

30.4.2 Software Standby Mode

(1) Transition to Software Standby Mode

The LSI switches from a program execution state to software standby mode by executing the SLEEP instruction when the STBY bit in STBCR is 1. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock output from the CK pin also halts.

The contents of the CPU registers and cache remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. Table 30.4 shows the states of peripheral module registers in software standby mode.

The CPU takes one cycle to finish writing to STBCR, and then executes processing for the next instruction. However, it takes one or more cycles to actually write. Therefore, execute a SLEEP instruction after reading STBCR to have the values written to STBCR by the CPU to be definitely reflected in the SLEEP instruction.

Table 30.4 Register States in Software Standby Mode

Module Name	Initialized Registers	Registers Whose Content is Retained
Interrupt controller (INTC)	—	All registers
Clock pulse generator (CPG)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
A/D converter (ADC)	All registers	—
I/O port	—	All registers
User debugging interface (H-UDI)	—	All registers
Serial communication interface with FIFO (SCIF)	—	All registers
Direct memory access controller (DMAC)	—	All registers
Multi-function timer pulse unit 2 (MTU2)	—	All registers
Multi-function timer pulse unit 2S (MTU2S)	—	All registers
Port output enable 2 (POE2)	—	All registers
Compare match timer (CMT)	—	All registers
I^2C bus interface 3 (IIC3)	BC[2:0] bits in ICMR register	Other than BC[2:0] bits in ICMR
Serial communication interface (SCI)	—	All registers
USB function module (USB)	—	All registers
Renesas serial peripheral interface (RSPI)	—	All registers
Controller area network (RCAN-IF)	—	All registers

The procedure for switching to software standby mode is as follows:

1. Clear the TME bit in the WDT's timer control register (WTCSR) to 0 to stop the WDT.
2. Set the WDT's timer counter (WTCNT) to 0 and the CKS[2:0] bits in WTCSR to appropriate values to secure the specified oscillation settling time.
3. After setting the STBY bit in STBCR to 1, read STBCR. Then, execute a SLEEP instruction.

(2) Exit from Software Standby Mode

Software standby mode is exited by interrupts (NMI and IRQ) and resets (a manual reset and power-on reset).

- Canceling with an interrupt

  When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) of the interrupt controller (INTC)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)) is detected, clock oscillation is started. This clock is supplied only to the oscillation settling counter (WDT).

  When the time, that has been specified in the clock select bits (CKS[2:0]) in the watchdog timer control/status register (WTCSR) of the WDT before the transition to the software standby mode, is elapsed, the WDT overflow is generated. This overflow starts to supply the clock to the entire LSI because it is used to decide that the clock is settled. Then, this releases the software standby mode and starts the NMI interrupt exception handling (IRQ interrupt exception handling for the IRRQ).

  To release the software standy mode by the NMI interrupt or IRQ interrupt, set bits CKS[2:0] so as the WDT overflow period is longer than the oscillation setting time.

  The clock output phase of the CK pin may be unstable immediately after detecting an interrupt and until software standby mode is released. When software standby mode is released by the falling edge of the NMI pin, the NMI pin should be high when the CPU enters software standby mode (when the clock pulse stops) and should be low when software standby mode is re-entered (when the clock is initiated after oscillation settling). When software standby mode is released by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters software standby mode (when the clock pulse stops) and should be high when software standby mode is re-entered (when the clock is initiated after oscillation settling). (The same applies to the IRQ pin.)

- Exit from software standby by a reset

  When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin is driven low, this LSI enters the power-on reset and manual reset and software standby mode is exited.

  Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation settles.

  Internal clock pulses are output continuously on the CK pin.

30.4.3 Application Example of Software Standy Mode

Figure 30.1 shows an example for the timing when software standy mode is entered at the falling edge of the NMI signal and released at the rising edge of the NMI signal.

When the NMI pin is changed from high to low while the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) is 0 (falling edge detection), an NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge selection) in the NMI exception service routine and the SLEEP instruction is executed with the STBY bit in STBCR is 1, the CPU enters the software standby mode. Then, software standby mode is released when the NMI pin is changed from low to high.

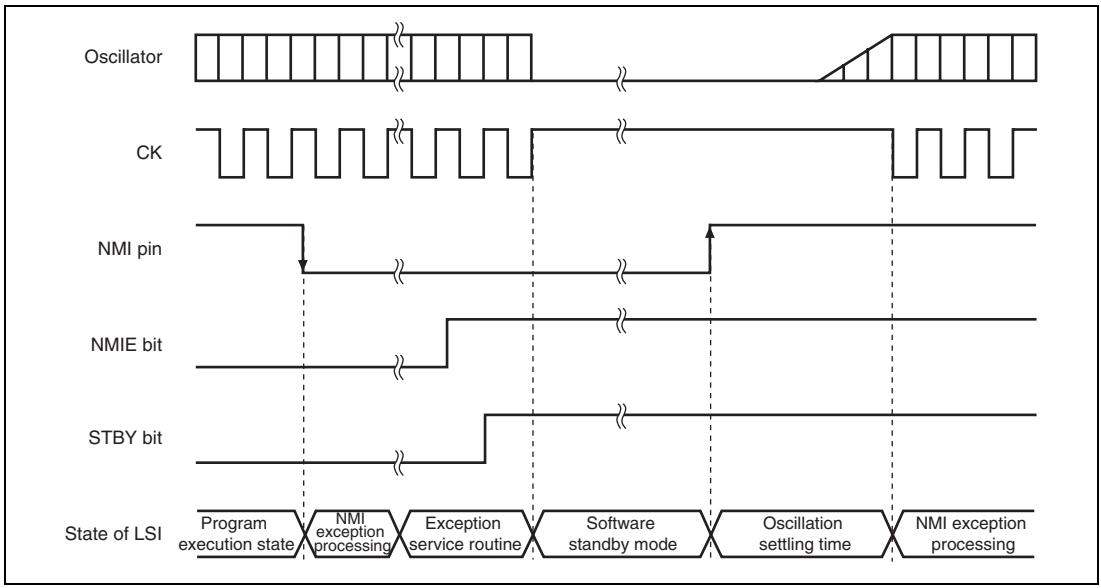


Figure 30.1 NMI Timing in Software Standby Mode (Application Example)

30.4.4 Module Standby Function

(1) Transition to Module Standby Function

Setting the standby control register MSTP bits to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in normal mode and sleep mode. Disable a module before placing it in module standby mode. In addition, do not access the module's registers while it is in the module standby state.

(2) Canceling Module Standby Function

The module standby function can be canceled by clearing the MSTP bits to 0, or by a power-on reset (only possible for H-UDI, UBC, DMAC, and DTC). When taking a module out of the module standby state by clearing the corresponding MSTP bit to 0, read the MSTP bit to confirm that it has been cleared to 0.

30.5 Usage Notes

30.5.1 Current Consumption during Oscillation Settling Time

While waiting for clock oscillation to settle, the current consumption is increased.

30.5.2 Notes on Writing to Registers

When writing to a register related to power-down modes by the CPU, after the CPU executes the write instruction, it then executes the subsequent instruction without waiting for the actual writing process to the register to finish.

To update the change made by writing to a register while executing the subsequent instruction, perform a dummy read to the same register between the instruction to write to the register and the subsequent instruction.

30.5.3 Notes on Canceling Software Standby Mode with an IRQx Interrupt Request

When canceling software standby mode using an IRQx interrupt request, change the IRQ sense select setting of ICRx in a state in which no IRQx interrupt requests are generated and clear the IRQxF flag in IRQRRx to 0 by the automatic clearing function of the IRQx interrupt processing.

If the IRQxF flag in the IRQ interrupt request register x (IRQRRx) is 1, changing the setting of the IRQ sense select bits in the interrupt control register x (ICRx) or clearing the IRQxF flag in IRQRRx to 0 will clear the relevant IRQx interrupt request but will not clear the software standby cancellation request.

Section 31 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for boundary scan function and emulator support.

This section mainly describes the boundary scan function. For the dedicated emulator function of the H-UDI, see the user's manual of the applicable emulator.

31.1 Features

The user debugging interface (H-UDI) is a serial input/output interface that conforms to IEEE 1149.1 and has boundary scan, reset, and H-UDI interrupt request functions.

When using an emulator, do not use this interface function.

For the method of connecting the emulator, see the emulator manual.

The H-UDI of this LSI provides the TAP controller for the boundary scan function, separately from the one for the other functions of the H-UDI. When the power is turned on, set the input to the $\overline{\text{ASEMD0}}$ pin to the high level and keep the $\overline{\text{TRST}}$ and $\overline{\text{RES}}$ pins asserted at the same time for a predetermined period of time, then the boundary scan TAP controller will be selected.

To use the reset or interrupt generation function, it is necessary to issue the switch to H-UDI command to the boundary scan TAP controller. The CPU cannot access the boundary scan control circuit.

Figure 31.1 shows a block diagram of the H-UDI.

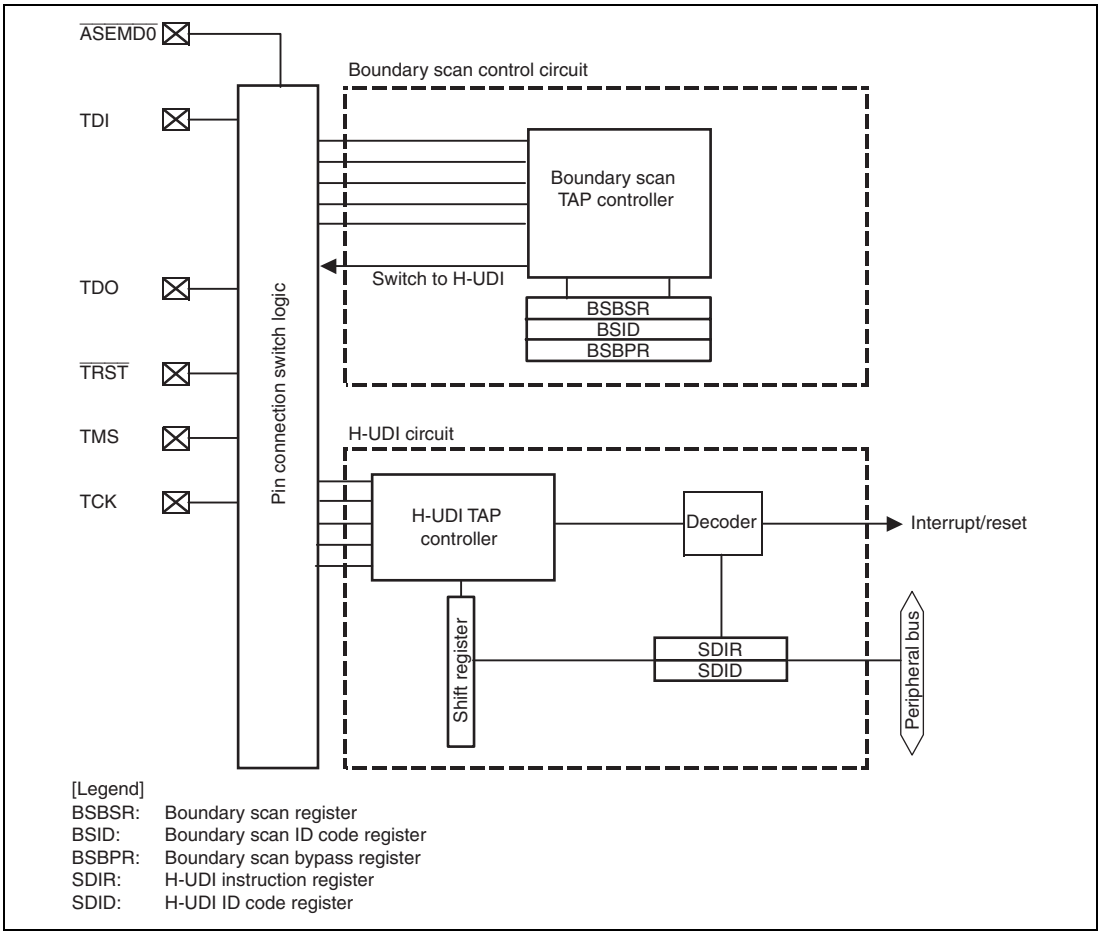


Figure 31.1 Block Diagram of H-UDI

31.2 Input/Output Pins

Table 31.1 lists the pins of the H-UDI.

Table 31.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial data input/output clock pin	TCK	Input	Data is serially supplied to the H-UDI from the data input pin (TDI), and output from the data output pin (TDO), in synchronization with this clock. This pin is pulled up within the chip.
Mode select input pin	TMS	Input	The state of the TAP controller is determined by changing this signal in synchronization with TCK. For the protocol, see figure 31.3. This pin is pulled up within the chip.
Reset input pin	$\overline{\text{TRST}}$	Input	Input is accepted asynchronously with respect to TCK, and when low, the boundary scan circuit and H-UDI are reset. $\overline{\text{TRST}}$ must be low for a predetermined period when power is turned on regardless of using the boundary scan circuit or H-UDI function. See section 31.6.2, Reset Configuration, for more information. This pin is pulled up within the chip.
Serial data input pin	TDI	Input	Data is input to the H-UDI at the rising edge of TCK. This pin is pulled up within the chip.
Serial data output pin	TDO	Output	Data is output from the H-UDI in synchronization with the falling edge of TCK.
ASE mode select input pin	$\overline{\text{ASEMD0}}$	Input	If a low level is input at the $\overline{\text{ASEMD0}}$ pin while the $\overline{\text{RES}}$ pin is asserted, ASE mode is entered; if a high level is input, product chip mode is entered. In ASE mode, the dedicated emulator function can be used. To use the boundary scan function, input a high level to $\overline{\text{ASEMD0}}$. Do not change the input level to $\overline{\text{ASEMD0}}$ unless the $\overline{\text{RES}}$ pin is asserted. This pin is pulled up within the chip.

31.3 Boundary Scan TAP Controller

The H-UDI of this LSI provides the TAP controller for the boundary scan function (hereafter referred to as the boundary scan TAP controller), separately from the one for the other functions of the H-UDI. When the power is turned on, set the input to $\overline{\text{ASEMD0}}$ to the high level and keep $\overline{\text{TRST}}$ and $\overline{\text{RES}}$ asserted at the same time for a predetermined period of time, then the boundary scan TAP controller will be selected and the boundary scan function will be enabled.

Note however that the following restrictions should be observed for this LSI.

1. The following pins are not subject to the boundary scan function.
   - Clock-related pins (EXTAL, XTAL, USBEXTAL, and USBXTAL)
   - USB-related pins (USD+ and USD-)
   - System-related pin ($\overline{\text{RES}}$)
   - H-UDI-related pins ($\overline{\text{TRST}}$, TMS, TCK, TDI, TDO, and $\overline{\text{ASEMD0}}$)
2. The PBI2 and PBI3 pins are provided with input boundary scan registers, but not with output (open drain output) boundary scan registers.
3. When the boundary scan function is executed, the maximum frequency of TCK is 6.25 MHz. When an H-UDI function is executed, the maximum frequency of TCK is 25 MHz.
4. When the power is turned on, input a low level to $\overline{\text{TRST}}$ at the same time with $\overline{\text{RES}}$ for a predetermined period of time and input the clock signal to EXTAL.
5. A transition to EXTEST, CLAMP, or HIGHZ resets the LSI. To make a transition to another mode from one of these, set $\overline{\text{ASEMD0}}$, FWE, MD1, and MD0 to the desired operation mode, input a low level to $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ at the same time for a predetermined period of time, and input the clock signal to EXTAL.
6. Even if a transition to HIGHZ is made, the $\overline{\text{WDTOVF}}$ pin is driven to the high level, but not at high impedance.

Table 31.2 lists the commands that the boundary scan TAP controller supports. If a command longer than 4 bits is issued from the TDI pin, the last 4 bits of the serial data become valid. Operation is not guaranteed if a reserved value defined in the table is input.

Figure 31.2 shows a switchover sequence from the boundary scan TAP controller to the H-UDI.

Table 31.2 Commands that Boundary Scan TAP Controller Supports

Bit 3	Bit 2	Bit 1	Bit 0	Description
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	0	1	0	CLAMP
0	0	1	1	HIGHZ
0	1	0	0	IDCODE (Initial value)
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Switch to H-UDI
1	1	1	1	BYPASS

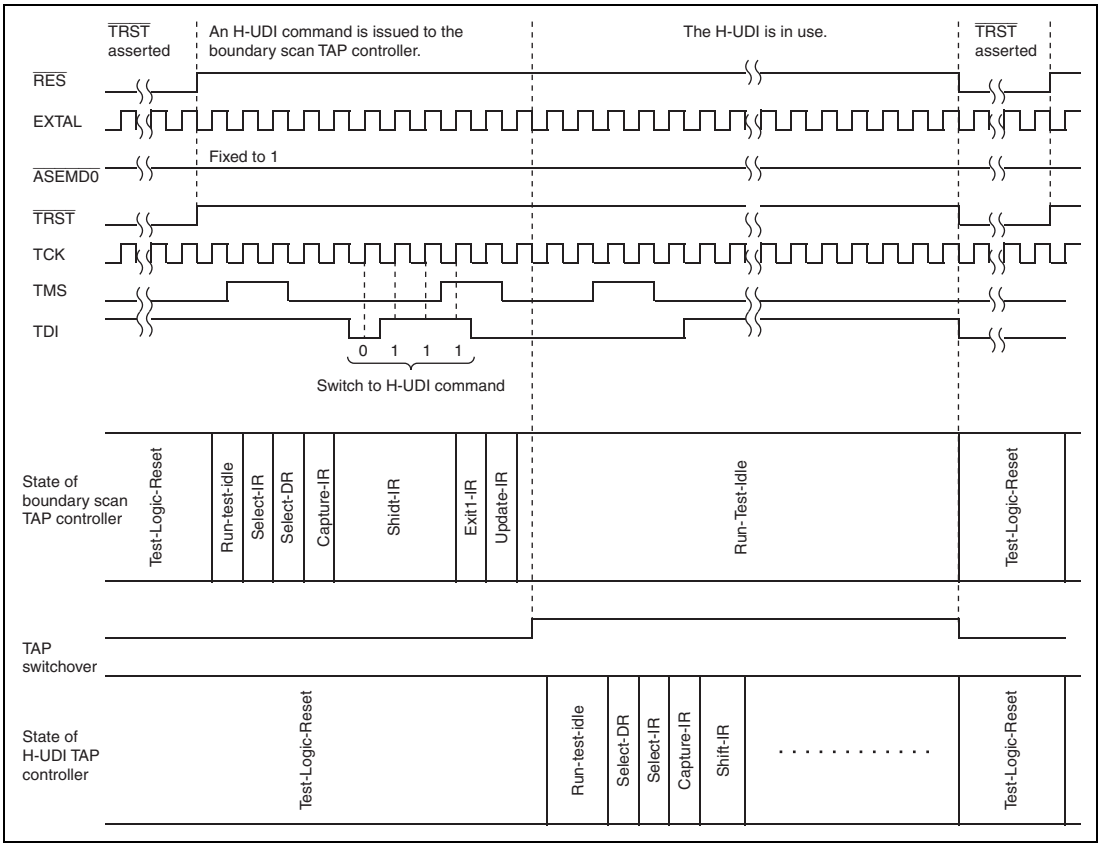


Figure 31.2 Switchover Sequence from Boundary Scan TAP Controller to H-UDI

31.4 H-UDI TAP Controller

The H-UDI of this LSI provides the TAP controller for the H-UDI functions (hereafter referred to as H-UDI TAP controller), separately from the boundary scan TAP controller.

This H-UDI TAP controller is enabled by issuing the switch to H-UDI command to the boundary scan TAP controller.

Table 31.3 lists the commands that the H-UDI TAP controller supports. If a command longer than 4 bits is issued from the TDI pin, the last 4 bits of the serial data become valid. Operation is not guaranteed if a reserved value defined in the table is input.

Table 31.3 Commands that H-UDI TAP Controller Supports

TI3	TI2	TI1	TI0	Description
0	0	0	0	Reserved
0	0	0	1	Reserved
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	H-UDI reset negate
0	1	1	1	H-UDI reset assert
1	0	0	0	Reserved
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	H-UDI interrupt
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	IDCODE (Initial value)
1	1	1	1	BYPASS

31.5 Register Descriptions

1. Registers of Boundary Scan Circuit

The boundary scan circuit has the following registers.

The on-chip CPU cannot access these registers.

- Bypass register (BSBPR)
- Instruction register (BSIR)
- ID register (BSID)
- Boundary scan register (BSBSR)

2. Registers of H-UDI Circuit

The H-UDI circuit has the following registers.

- Instruction register (SDIR)
- ID register (SDID)

31.5.1 Bypass Register (BSBPR)

BSBPR of the boundary scan circuit is a 1-bit register. When the BYPASS command is set, BSBPR is connected between the TDI and TDO pins. The initial value is undefined.

31.5.2 Instruction Register (BSIR)

BSIR of the boundary scan circuit is a 4-bit register that stores a command for the boundary scan TAP controller. The commands that this LSI supports are listed in table 31.2. The initial value of this register is IDCODE (4'b0100). SDIR is initialized when $\overline{\text{TRST}}$ is low or when in the TAP test-logic-reset state, and can be written to by using the pins listed in table 31.1 irrespective of CPU operation. When a command longer than 4 bits is issued from the TDI pin, the last 4 bits of the serial data are stored in this register. Operation is not guaranteed if a reserved value is set in this register.

31.5.3 ID Register (BSID)

BSID of the boundary scan circuit stores the ID code of this LSI (H'08083447). Set the IDCODE command in the boundary scan circuit and set the TAP state to Shift-DR, then this value can be read from the TDO pin.

31.5.4 Boundary Scan Register (BSBSR)

BSBSR of the boundary scan circuit is a shift register arranged on a pad and controls external input/output pins.

Using the commands listed in table 31.2, a boundary scan test conforming to the JTAG standard can be performed.

This register cannot be initialized.

Table 31.4 Boundary Scan Registers

No.	Pin Name	Type
	From TDI	
321	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	OUTPUT
320	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	CONTROL
319	FWE/$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	INPUT
318	PF0/AN0	INPUT
317	PF1/AN1	INPUT
316	PF2/AN2	INPUT
315	PF3/AN3	INPUT
314	PF4/AN4	INPUT
313	PF5/AN5	INPUT
312	PF6/AN6	INPUT
311	PF7/AN7	INPUT
310	MD0	INPUT
309	MD1	INPUT
308	$\overline{\text{WDTOVF}}$	OUTPUT
307	$\overline{\text{WDTOVF}}$	CONTROL
306	—	INTERNAL
305	PA0/$\overline{\text{CS0}}$/IRQ4/CRx0/RXD0/RX_CLK	OUTPUT
304	PA0/$\overline{\text{CS0}}$/IRQ4/CRx0/RXD0/RX_CLK	CONTROL
303	PA0/$\overline{\text{CS0}}$/IRQ4/CRx0/RXD0/RX_CLK	INPUT
302	PA1/$\overline{\text{CS1}}$/IRQ5/CTx0/TXD0/MII_RXD0	OUTPUT
301	PA1/$\overline{\text{CS1}}$/IRQ5/CTx0/TXD0/MII_RXD0	CONTROL
300	PA1/$\overline{\text{CS1}}$/IRQ5/CTx0/TXD0/MII_RXD0	INPUT

No.	Pin Name	Type
299	PA2/$\overline{CS2}$/TCLKD/SSL0/SCK0/MII_RXD1	OUTPUT
298	PA2/$\overline{CS2}$/TCLKD/SSL0/SCK0/MII_RXD1	CONTROL
297	PA2/$\overline{CS2}$/TCLKD/SSL0/SCK0/MII_RXD1	INPUT
296	PA3/$\overline{CS3}$/TCLKC/MISO/RXD1/MII_RXD2	OUTPUT
295	PA3/$\overline{CS3}$/TCLKC/MISO/RXD1/MII_RXD2	CONTROL
294	PA3/$\overline{CS3}$/TCLKC/MISO/RXD1/MII_RXD2	INPUT
293	PA4/$\overline{CS4}$/TCLKB/MOSI/TXD1/MII_RXD3	OUTPUT
292	PA4/$\overline{CS4}$/TCLKB/MOSI/TXD1/MII_RXD3	CONTROL
291	PA4/$\overline{CS4}$/TCLKB/MOSI/TXD1/MII_RXD3	INPUT
290	PA5/$\overline{CS5}$/TCLKA/RSPCK/SCK1/RX_ER	OUTPUT
289	PA5/$\overline{CS5}$/TCLKA/RSPCK/SCK1/RX_ER	CONTROL
288	PA5/$\overline{CS5}$/TCLKA/RSPCK/SCK1/RX_ER	INPUT
287	PE7/$\overline{UBCTRG}$/TIOC2B/SSL1/RXD2/RX_DV	OUTPUT
286	PE7/$\overline{UBCTRG}$/TIOC2B/SSL1/RXD2/RX_DV	CONTROL
285	PE7/$\overline{UBCTRG}$/TIOC2B/SSL1/RXD2/RX_DV	INPUT
284	PE8/DREQ2/TIOC3A/SSL2/SCK2/EXOUT	OUTPUT
283	PE8/DREQ2/TIOC3A/SSL2/SCK2/EXOUT	CONTROL
282	PE8/DREQ2/TIOC3A/SSL2/SCK2/EXOUT	INPUT
281	PE10/DREQ3/TIOC3C/SSL3/TXD2/TX_CLK	OUTPUT
280	PE10/DREQ3/TIOC3C/SSL3/TXD2/TX_CLK	CONTROL
279	PE10/DREQ3/TIOC3C/SSL3/TXD2/TX_CLK	INPUT
278	PE9/DACK2/TIOC3B/TX_EN	OUTPUT
277	PE9/DACK2/TIOC3B/TX_EN	CONTROL
276	PE9/DACK2/TIOC3B/TX_EN	INPUT
275	PE11/DACK3/TIOC3D/MII_TXD0	OUTPUT
274	PE11/DACK3/TIOC3D/MII_TXD0	CONTROL
273	PE11/DACK3/TIOC3D/MII_TXD0	INPUT
272	PE12/TIOC4A/MII_TXD1	OUTPUT
271	PE12/TIOC4A/MII_TXD1	CONTROL
270	PE12/TIOC4A/MII_TXD1	INPUT
269	PE13/$\overline{MRES}$/TIOC4B/MII_TXD2	OUTPUT
268	PE13/$\overline{MRES}$/TIOC4B/MII_TXD2	CONTROL

No.	Pin Name	Type
267	PE13/$\overline{\text{MRES}}$/TIOC4B/MII_TXD	INPUT
266	PE14/DACK0/TIOC4C/MII_TXD3	OUTPUT
265	PE14/DACK0/TIOC4C/MII_TXD3	CONTROL
264	PE14/DACK0/TIOC4C/MII_TXD3	INPUT
263	PE15/DACK1/TIOC4D/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/TX_ER	OUTPUT
262	PE15/DACK1/TIOC4D/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/TX_ER	CONTROL
261	PE15/DACK1/TIOC4D/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/TX_ER	INPUT
260	PE0/DREQ0/TIOC0A/TIOC4AS/LNKSTA	OUTPUT
259	PE0/DREQ0/TIOC0A/TIOC4AS/LNKSTA	CONTROL
258	PE0/DREQ0/TIOC0A/TIOC4AS/LNKSTA	INPUT
257	PE1/TEND0/TIOC0B/TIOC4BS/MDC	OUTPUT
256	PE1/TEND0/TIOC0B/TIOC4BS/MDC	CONTROL
255	PE1/TEND0/TIOC0B/TIOC4BS/MDC	INPUT
254	PE2/DREQ1/TIOC0C/TIOC4CS/WOL	OUTPUT
253	PE2/DREQ1/TIOC0C/TIOC4CS/WOL	CONTROL
252	PE2/DREQ1/TIOC0C/TIOC4CS/WOL	INPUT
251	PE3/TEND1/TIOC0D/TIOC4DS/COL	OUTPUT
250	PE3/TEND1/TIOC0D/TIOC4DS/COL	CONTROL
249	PE3/TEND1/TIOC0D/TIOC4DS/COL	INPUT
248	PE4/IRQ4/TIOC1A/$\overline{\text{POE8}}$/SCK3/CRS	OUTPUT
247	PE4/IRQ4/TIOC1A/$\overline{\text{POE8}}$/SCK3/CRS	CONTROL
246	PE4/IRQ4/TIOC1A/$\overline{\text{POE8}}$/SCK3/CRS	INPUT
245	PE5/TIOC1B/TIOC3BS/TXD3/MDIO	OUTPUT
244	PE5/TIOC1B/TIOC3BS/TXD3/MDIO	CONTROL
243	PE5/TIOC1B/TIOC3BS/TXD3/MDIO	INPUT
242	PE6/TIOC2A/TIOC3DS/RXD3	OUTPUT
241	PE6/TIOC2A/TIOC3DS/RXD3	CONTROL
240	PE6/TIOC2A/TIOC3DS/RXD3	INPUT
239	PA21/$\overline{\text{RD}}$/$\overline{\text{BACK}}$/IRQ5/CKE/$\overline{\text{POE3}}$/SCK1/$\overline{\text{FRAME}}$	OUTPUT
238	PA21/$\overline{\text{RD}}$/$\overline{\text{BACK}}$/IRQ5/CKE/$\overline{\text{POE3}}$/SCK1/$\overline{\text{FRAME}}$	CONTROL
237	PA21/$\overline{\text{RD}}$/$\overline{\text{BACK}}$/IRQ5/CKE/$\overline{\text{POE3}}$/SCK1/$\overline{\text{FRAME}}$	INPUT
236	PA20/$\overline{\text{WRL}}$/DQMLL/$\overline{\text{BREQ}}$/IRQ6/$\overline{\text{CASU}}$/$\overline{\text{POE4}}$/TXD1/$\overline{\text{AH}}$	OUTPUT

No.	Pin Name	Type
235	PA20/$\overline{\text{WRL}}$/DQMLL/$\overline{\text{BREQ}}$/IRQ6/$\overline{\text{CASU}}$/POE4/TXD1/$\overline{\text{AH}}$	CONTROL
234	PA20/$\overline{\text{WRL}}$/DQMLL/$\overline{\text{BREQ}}$/IRQ6/$\overline{\text{CASU}}$/POE4/TXD1/$\overline{\text{AH}}$	INPUT
233	PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$	OUTPUT
232	PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$	CONTROL
231	PA19/$\overline{\text{WRH}}$/DQMLU/$\overline{\text{WAIT}}$/IRQ7/$\overline{\text{RASU}}$/$\overline{\text{POE8}}$/RXD1/$\overline{\text{BS}}$	INPUT
230	PA18/CK	OUTPUT
229	PA18/CK	CONTROL
228	PA18/CK	INPUT
227	PA17/$\overline{\text{RD}}$	OUTPUT
226	PA17/$\overline{\text{RD}}$	CONTROL
225	PA17/$\overline{\text{RD}}$	INPUT
224	PA16/$\overline{\text{WRL}}$/DQMLL	OUTPUT
223	PA16/$\overline{\text{WRL}}$/DQMLL	CONTROL
222	PA16/$\overline{\text{WRL}}$/DQMLL	INPUT
221	PA15/$\overline{\text{WRH}}$/DQMLU	OUTPUT
220	PA15/$\overline{\text{WRH}}$/DQMLU	CONTROL
219	PA15/$\overline{\text{WRH}}$/DQMLU	INPUT
218	PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$	OUTPUT
217	PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$	CONTROL
216	PA14/$\overline{\text{WRHH}}$/DQMUU/$\overline{\text{RASL}}$	INPUT
215	PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$	OUTPUT
214	PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$	CONTROL
213	PA13/$\overline{\text{WRHL}}$/DQMUL/$\overline{\text{CASL}}$	INPUT
212	PC0/A0/IRQ4/$\overline{\text{POE0}}$	OUTPUT
211	PC0/A0/IRQ4/$\overline{\text{POE0}}$	CONTROL
210	PC0/A0/IRQ4/$\overline{\text{POE0}}$	INPUT
209	PC1/A1	OUTPUT
208	PC1/A1	CONTROL
207	PC1/A1	INPUT
206	PC2/A2	OUTPUT
205	PC2/A2	CONTROL
204	PC2/A2	INPUT

No.	Pin Name	Type
203	PC3/A3	OUTPUT
202	PC3/A3	CONTROL
201	PC3/A3	INPUT
200	PC4/A4	OUTPUT
199	PC4/A4	CONTROL
198	PC4/A4	INPUT
197	PC5/A5	OUTPUT
196	PC5/A5	CONTROL
195	PC5/A5	INPUT
194	PC6/A6	OUTPUT
193	PC6/A6	CONTROL
192	PC6/A6	INPUT
191	PC7/A7	OUTPUT
190	PC7/A7	CONTROL
189	PC7/A7	INPUT
188	PC8/A8/CRx0/RXD0	OUTPUT
187	PC8/A8/CRx0/RXD0	CONTROL
186	PC8/A8/CRx0/RXD0	INPUT
185	PC9/A9/CTx0/TXD0	OUTPUT
184	PC9/A9/CTx0/TXD0	CONTROL
183	PC9/A9/CTx0/TXD0	INPUT
182	PC10/A10/TIOC1A/CRx0/RXD0	OUTPUT
181	PC10/A10/TIOC1A/CRx0/RXD0	CONTROL
180	PC10/A10/TIOC1A/CRx0/RXD0	INPUT
179	PC11/A11/TIOC1B/CTx0/TXD0	OUTPUT
178	PC11/A11/TIOC1B/CTx0/TXD0	CONTROL
177	PC11/A11/TIOC1B/CTx0/TXD0	INPUT
176	PC12/A12/TCLKA	OUTPUT
175	PC12/A12/TCLKA	CONTROL
174	PC12/A12/TCLKA	INPUT
173	PC13/A13/IRQ0/TCLKB	OUTPUT
172	PC13/A13/IRQ0/TCLKB	CONTROL

No.	Pin Name	Type
171	PC13/A13/IRQ0/TCLKB	INPUT
170	PC14/A14/IRQ1/TCLKC	OUTPUT
169	PC14/A14/IRQ1/TCLKC	CONTROL
168	PC14/A14/IRQ1/TCLKC	INPUT
167	PC15/A15/IRQ2/TCLKD	OUTPUT
166	PC15/A15/IRQ2/TCLKD	CONTROL
165	PC15/A15/IRQ2/TCLKD	INPUT
164	PB0/A16/RD/$\overline{WR}$/IRQ0/TIOC2A	OUTPUT
163	PB0/A16/RD/$\overline{WR}$/IRQ0/TIOC2A	CONTROL
162	PB0/A16/RD/$\overline{WR}$/IRQ0/TIOC2A	INPUT
161	PB1/A17/$\overline{IRQOUT}$/$\overline{REFOUT}$/IRQ1/TIOC0A/$\overline{ADTRG}$	OUTPUT
160	PB1/A17/$\overline{IRQOUT}$/$\overline{REFOUT}$/IRQ1/TIOC0A/$\overline{ADTRG}$	CONTROL
159	PB1/A17/$\overline{IRQOUT}$/$\overline{REFOUT}$/IRQ1/TIOC0A/$\overline{ADTRG}$	INPUT
158	PB2/A18/$\overline{BACK}$/IRQ2/TIOC0B/$\overline{RASL}$/RXD3/$\overline{FRAME}$	OUTPUT
157	PB2/A18/$\overline{BACK}$/IRQ2/TIOC0B/$\overline{RASL}$/RXD3/$\overline{FRAME}$	CONTROL
156	PB2/A18/$\overline{BACK}$/IRQ2/TIOC0B/$\overline{RASL}$/RXD3/$\overline{FRAME}$	INPUT
155	PB3/A19/$\overline{BREQ}$/IRQ3/TIOC0C/$\overline{CASL}$/TXD3/$\overline{AH}$	OUTPUT
154	PB3/A19/$\overline{BREQ}$/IRQ3/TIOC0C/$\overline{CASL}$/TXD3/$\overline{AH}$	CONTROL
153	PB3/A19/$\overline{BREQ}$/IRQ3/TIOC0C/$\overline{CASL}$/TXD3/$\overline{AH}$	INPUT
152	PB4/A20/$\overline{BACK}$/IRQ4/TIOC0D/$\overline{WAIT}$/SCK3/$\overline{BS}$	OUTPUT
151	PB4/A20/$\overline{BACK}$/IRQ4/TIOC0D/$\overline{WAIT}$/SCK3/$\overline{BS}$	CONTROL
150	PB4/A20/$\overline{BACK}$/IRQ4/TIOC0D/$\overline{WAIT}$/SCK3/$\overline{BS}$	INPUT
149	PB5/A21/$\overline{BREQ}$/IRQ5/RXD0	OUTPUT
148	PB5/A21/$\overline{BREQ}$/IRQ5/RXD0	CONTROL
147	PB5/A21/$\overline{BREQ}$/IRQ5/RXD0	INPUT
146	PB6/A22/$\overline{WAIT}$/IRQ6/TCLKD/TXD0	OUTPUT
145	PB6/A22/$\overline{WAIT}$/IRQ6/TCLKD/TXD0	CONTROL
144	PB6/A22/$\overline{WAIT}$/IRQ6/TCLKD/TXD0	INPUT
143	PB7/A23/TEND0/IRQ7/TCLKC/SCK4/RD/$\overline{WR}$	OUTPUT
142	PB7/A23/TEND0/IRQ7/TCLKC/SCK4	CONTROL
141	PB7/A23/TEND0/IRQ7/TCLKC/SCK4	INPUT
140	PB8/A24/DREQ0/TCLKB/RXD4/$\overline{CS2}$	OUTPUT

No.	Pin Name	Type
139	PB8/A24/DREQ0/TCLKB/RXD4/$\overline{CS2}$	CONTROL
138	PB8/A24/DREQ0/TCLKB/RXD4/$\overline{CS2}$	INPUT
137	PB9/A25/DACK0/TCLKA/TXD4/$\overline{CS3}$	OUTPUT
136	PB9/A25/DACK0/TCLKA/TXD4/$\overline{CS3}$	CONTROL
135	PB9/A25/DACK0/TCLKA/TXD4/$\overline{CS3}$	INPUT
134	PB10/$\overline{CS0}$/$\overline{CS2}$/IRQ0/RXD2/$\overline{CS6}$	OUTPUT
133	PB10/$\overline{CS0}$/$\overline{CS2}$/IRQ0/RXD2/$\overline{CS6}$	CONTROL
132	PB10/$\overline{CS0}$/$\overline{CS2}$/IRQ0/RXD2/$\overline{CS6}$	INPUT
131	PB11/$\overline{CS1}$/$\overline{CS3}$/IRQ1/TXD2/$\overline{CS7}$	OUTPUT
130	PB11/$\overline{CS1}$/$\overline{CS3}$/IRQ1/TXD2/$\overline{CS7}$	CONTROL
129	PB11/$\overline{CS1}$/$\overline{CS3}$/IRQ1/TXD2/$\overline{CS7}$	INPUT
128	PD0/D0	OUTPUT
127	PD0/D0	CONTROL
126	PD0/D0	INPUT
125	PD1/D1	OUTPUT
124	PD1/D1	CONTROL
123	PD1/D1	INPUT
122	PD2/D2/TIC5U/RXD2	OUTPUT
121	PD2/D2/TIC5U/RXD2	CONTROL
120	PD2/D2/TIC5U/RXD2	INPUT
119	PD3/D3/TIC5V/TXD2	OUTPUT
118	PD3/D3/TIC5V/TXD2	CONTROL
117	PD3/D3/TIC5V/TXD2	INPUT
116	PD4/D4/TIC5W/SCK2	OUTPUT
115	PD4/D4/TIC5W/SCK2	CONTROL
114	PD4/D4/TIC5W/SCK2	INPUT
113	PD5/D5/TIC5US	OUTPUT
112	PD5/D5/TIC5US	CONTROL
111	PD5/D5/TIC5US	INPUT
110	PD6/D6/TIC5VS	OUTPUT
109	PD6/D6/TIC5VS	CONTROL
108	PD6/D6/TIC5VS	INPUT

No.	Pin Name	Type
107	PD7/D7/TIC5WS	OUTPUT
106	PD7/D7/TIC5WS	CONTROL
105	PD7/D7/TIC5WS	INPUT
104	PD8/D8/TIOC3AS	OUTPUT
103	PD8/D8/TIOC3AS	CONTROL
102	PD8/D8/TIOC3AS	INPUT
101	PD9/D9/TIOC3CS	OUTPUT
100	PD9/D9/TIOC3CS	CONTROL
99	PD9/D9/TIOC3CS	INPUT
98	PD10/D10/TIOC3BS	OUTPUT
97	PD10/D10/TIOC3BS	CONTROL
96	PD10/D10/TIOC3BS	INPUT
95	PD11/D11/TIOC3DS	OUTPUT
94	PD11/D11/TIOC3DS	CONTROL
93	PD11/D11/TIOC3DS	INPUT
92	PD12/D12/TIOC4AS	OUTPUT
91	PD12/D12/TIOC4AS	CONTROL
90	PD12/D12/TIOC4AS	INPUT
89	PD13/D13/AUDCK/TIOC4BS	OUTPUT
88	PD13/D13/AUDCK/TIOC4BS	CONTROL
87	PD13/D13/AUDCK/TIOC4BS	INPUT
86	PD14/D14/TIOC4CS	OUTPUT
85	PD14/D14/TIOC4CS	CONTROL
84	PD14/D14/TIOC4CS	INPUT
83	PD15/D15/TIOC4DS	OUTPUT
82	PD15/D15/TIOC4DS	CONTROL
81	PD15/D15/TIOC4DS	INPUT
80	PD16/D16/$\overline{\text{UBCTRG}}$/IRQ0/AUDATA0/$\overline{\text{POE0}}$	OUTPUT
79	PD16/D16/$\overline{\text{UBCTRG}}$/IRQ0/AUDATA0/$\overline{\text{POE0}}$	CONTROL
78	PD16/D16/$\overline{\text{UBCTRG}}$/IRQ0/AUDATA0/$\overline{\text{POE0}}$	INPUT
77	PD17/D17/IRQ1/AUDATA1/$\overline{\text{POE4}}$/$\overline{\text{ADTRG}}$	OUTPUT
76	PD17/D17/IRQ1/AUDATA1/$\overline{\text{POE4}}$/$\overline{\text{ADTRG}}$	CONTROL

No.	Pin Name	Type
75	PD17/D17/IRQ1/AUDATA1/$\overline{\text{POE4}}$/$\overline{\text{ADTRG}}$	INPUT
74	PD18/D18/IRQ2/AUDATA2/MDIO	OUTPUT
73	PD18/D18/IRQ2/AUDATA2/MDIO	CONTROL
72	PD18/D18/IRQ2/AUDATA2/MDIO	INPUT
71	PD19/D19/IRQ3/AUDATA3/LNKSTA	OUTPUT
70	PD19/D19/IRQ3/AUDATA3/LNKSTA	CONTROL
69	PD19/D19/IRQ3/AUDATA3/LNKSTA	INPUT
68	PD20/D20/IRQ4/$\overline{\text{AUDSYNC}}$/MDC	OUTPUT
67	PD20/D20/IRQ4/$\overline{\text{AUDSYNC}}$/MDC	CONTROL
66	PD20/D20/IRQ4/$\overline{\text{AUDSYNC}}$/MDC	INPUT
65	PD21/D21/TEND1/IRQ5/AUDCK/EXOUT	OUTPUT
64	PD21/D21/TEND1/IRQ5/AUDCK/EXOUT	CONTROL
63	PD21/D21/TEND1/IRQ5/AUDCK/EXOUT	INPUT
62	PD22/D22/DREQ1/IRQ6/WOL	OUTPUT
61	PD22/D22/DREQ1/IRQ6/WOL	CONTROL
60	PD22/D22/DREQ1/IRQ6/WOL	INPUT
59	PD23/D23/DACK1/IRQ7/COL	OUTPUT
58	PD23/D23/DACK1/IRQ7/COL	CONTROL
57	PD23/D23/DACK1/IRQ7/COL	INPUT
56	PD24/D24/TIOC4DS/CRS	OUTPUT
55	PD24/D24/TIOC4DS/CRS	CONTROL
54	PD24/D24/TIOC4DS/CRS	INPUT
53	PD25/D25/TIOC4CS/RX_CLK	OUTPUT
52	PD25/D25/TIOC4CS/RX_CLK	CONTROL
51	PD25/D25/TIOC4CS/RX_CLK	INPUT
50	PD26/D26/TIOC4BS/MII_RXD0	OUTPUT
49	PD26/D26/TIOC4BS/MII_RXD0	CONTROL
48	PD26/D26/TIOC4BS/MII_RXD0	INPUT
47	PD27/D27/TIOC4AS/MII_RXD1	OUTPUT
46	PD27/D27/TIOC4AS/MII_RXD1	CONTROL
45	PD27/D27/TIOC4AS/MII_RXD1	INPUT
44	PD28/D28/TIOC3DS/MII_RXD2	OUTPUT

No.	Pin Name	Type
43	PD28/D28/TIOC3DS/MII_RXD2	CONTROL
42	PD28/D28/TIOC3DS/MII_RXD2	INPUT
41	PD29/D29/TIOC3BS/MII_RXD3	OUTPUT
40	PD29/D29/TIOC3BS/MII_RXD3	CONTROL
39	PD29/D29/TIOC3BS/MII_RXD3	INPUT
38	PD30/D30/TIOC3CS/SSL3/RX_ER	OUTPUT
37	PD30/D30/TIOC3CS/SSL3/RX_ER	CONTROL
36	PD30/D30/TIOC3CS/SSL3/RX_ER	INPUT
35	PD31/D31/TIOC3AS/SSL2/RX_DV	OUTPUT
34	PD31/D31/TIOC3AS/SSL2/RX_DV	CONTROL
33	PD31/D31/TIOC3AS/SSL2/RX_DV	INPUT
32	PA12/$\overline{\text{CS0}}$/IRQ0/TIC5U/SSL1/TX_CLK	OUTPUT
31	PA12/$\overline{\text{CS0}}$/IRQ0/TIC5U/SSL1/TX_CLK	CONTROL
30	PA12/$\overline{\text{CS0}}$/IRQ0/TIC5U/SSL1/TX_CLK	INPUT
29	PA11/$\overline{\text{CS1}}$/IRQ1/TIC5V/CRx0/RXD0/TX_EN	OUTPUT
28	PA11/$\overline{\text{CS1}}$/IRQ1/TIC5V/CRx0/RXD0/TX_EN	CONTROL
27	PA11/$\overline{\text{CS1}}$/IRQ1/TIC5V/CRx0/RXD0/TX_EN	INPUT
26	PA10/$\overline{\text{CS2}}$/IRQ2/TIC5W/CTx0/TXD0/MII_TXD0	OUTPUT
25	PA10/$\overline{\text{CS2}}$/IRQ2/TIC5W/CTx0/TXD0/MII_TXD0	CONTROL
24	PA10/$\overline{\text{CS2}}$/IRQ2/TIC5W/CTx0/TXD0/MII_TXD0	INPUT
23	PA9/$\overline{\text{CS3}}$/IRQ3/TCLKD/SSL0/SCK0/MII_TXD1	OUTPUT
22	PA9/$\overline{\text{CS3}}$/IRQ3/TCLKD/SSL0/SCK0/MII_TXD1	CONTROL
21	PA9/$\overline{\text{CS3}}$/IRQ3/TCLKD/SSL0/SCK0/MII_TXD1	INPUT
20	PA8/$\overline{\text{CS4}}$/IRQ4/TCLKC/MISO/RXD1/MII_TXD2	OUTPUT
19	PA8/$\overline{\text{CS4}}$/IRQ4/TCLKC/MISO/RXD1/MII_TXD2	CONTROL
18	PA8/$\overline{\text{CS4}}$/IRQ4/TCLKC/MISO/RXD1/MII_TXD2	INPUT
17	PA7/$\overline{\text{CS5}}$/IRQ5/TCLKB/MOSI/TXD1/MII_TXD3	OUTPUT
16	PA7/$\overline{\text{CS5}}$/IRQ5/TCLKB/MOSI/TXD1/MII_TXD3	CONTROL
15	PA7/$\overline{\text{CS5}}$/IRQ5/TCLKB/MOSI/TXD1/MII_TXD3	INPUT
14	PA6/$\overline{\text{CS6}}$/IRQ6/TCLKA/RSPCK/SCK1/TX_ER	OUTPUT
13	PA6/$\overline{\text{CS6}}$/IRQ6/TCLKA/RSPCK/SCK1/TX_ER	CONTROL
12	PA6/$\overline{\text{CS6}}$/IRQ6/TCLKA/RSPCK/SCK1/TX_ER	INPUT

No.	Pin Name	Type
11	PB12/IRQ2/$\overline{\text{POE1}}$/SCL	INPUT
10	PB13/IRQ3/$\overline{\text{POE2}}$/SDA	INPUT
9	PB14/IRQ6	OUTPUT
8	PB14/IRQ6	CONTROL
7	PB14/IRQ6	INPUT
6	PB15/IRQ7	OUTPUT
5	PB15/IRQ7	CONTROL
4	PB15/IRQ7	INPUT
3	VBUS	OUTPUT
2	VBUS	CONTROL
1	VBUS	INPUT
0	NMI	INPUT
	To TDO	

31.5.5 Instruction Register (SDIR)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	T13	T12	T11	T10	–	–	–	–	–	–	–	–	–	–	–	–
Initial value:	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

SDIR of the H-UDI circuit is a 16-bit register that stores a command for the H-UDI TAP controller.

This register is allocated to the address of H'FFFE2000. This register can be read by the CPU, but cannot be written by the CPU.

The initial value of this register is IDCODE (16'hEFFD). It is initialized when $\overline{\text{TRST}}$ is low or when in the TAP test-logic-reset state, and can be written by input from the pins listed in table 31.1 after setting the switch to H-UDI command in the boundary scan TAP controller.

When a command longer than 4 bits is issued from the TDI pin, the last 4 bits of the serial data are stored in this register. Operation is not guaranteed if a reserved value is set in this register.

31.5.6 ID Register (SDID)

SDID of the H-UDI circuit stores the ID code of this LSI (H'08083447). Just as in the case of the boundary scan circuit, set the IDCODE command in the H-UDI TAP controller and set the TAP state to Shift-DR, then this value can be read from the TDO pin.

31.6 Operation

31.6.1 TAP Controller

Figure 31.3 shows the internal states of the TAP controller.

- The transition condition is the TMS value at the rising edge of TCK.
- The TDI value is sampled at the rising edge of TCK.
- The TDO value changes at the falling edge of TCK. The TDO value is at high impedance, except with Shift-DR and Shift-IR states.
- When a low level is input to $\overline{\text{TRST}}$, a transition to the test-logic-rest state occurs.

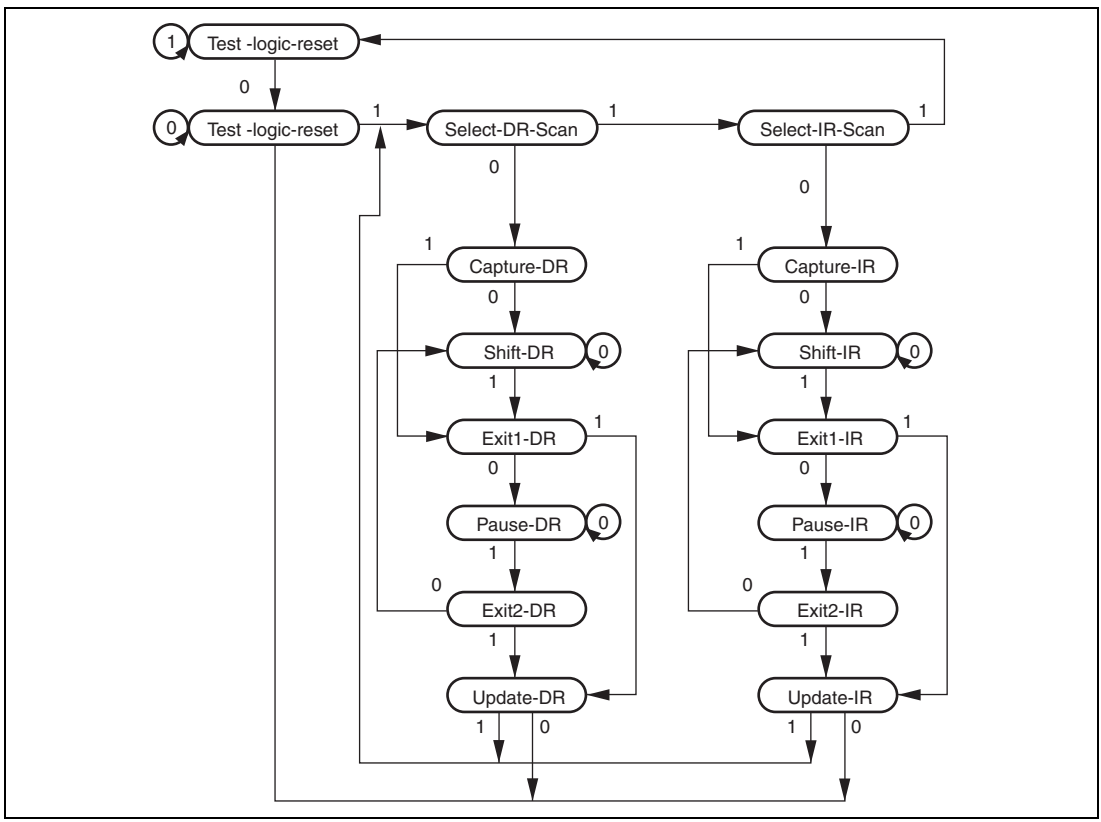


Figure 31.3 Internal States of TAP Controller

31.6.2 Reset Configuration

Table 31.5 shows the reset configuration of this chip.

Table 31.5 Reset Configuration

Operation Mode	$\overline{\text{ASEMD0}}$	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State
Product chip mode	1	0	0	Power-on reset and H-UDI reset
			1	Power-on reset
		1	0	H-UDI reset only (Normal operation)
			1	Normal operation
ASE mode*1	0	0	0	Power-on reset and H-UDI reset*2
			1	Power-on reset
		1	0	H-UDI reset only
			1	Normal operation

Notes: 1. ASE mode is used for emulator connection. In this mode, the boundary scan and H-UDI functions cannot be used.
2. Reset hold is entered if the $\overline{\text{TRST}}$ pin in driven low while the $\overline{\text{RES}}$ pin is negated. In this state, the CPU does not start up.

31.6.3 H-UDI Reset

An H-UDI reset is executed by setting an H-UDI reset assert command in the H-UDI TAP controller after setting the switch to H-UDI command (see figure 31.4). An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by setting the H-UDI reset negate command.

The required time between the H-UDI reset assert command and H-UDI reset negate command is the same as the time for keeping the $\overline{\text{RES}}$ pin low to apply a power-on reset.

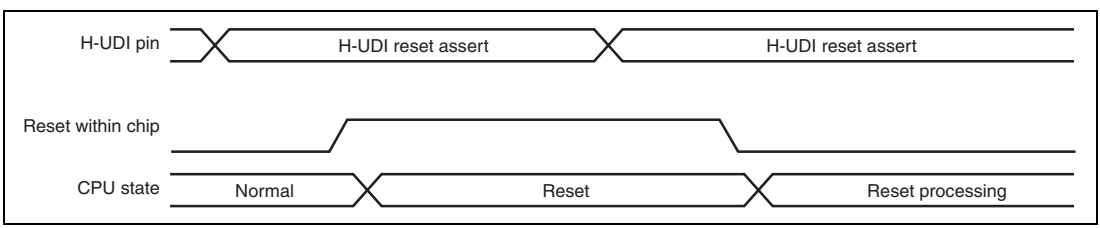


Figure 31.4 H-UDI Reset

31.6.4 H-UDI Interrupt

An interrupt is generated when the H-UDI interrupt command is set in the H-UDI TAP controller after the switch to H-UDI command is set. An H-UDI interrupt has a priority level of 15 and vector number of 14, and jumps to an address based on VBR and returns with an RTE instruction.

31.6.5 Boundary Scan Operation

This LSI supports the following commands: BYPASS, SAMPLE/PRELOAD, EXTEST, CLAMP, HIGHZ, and IDCODE.

(1) BYPASS

The BYPASS command is a mandatory and standard instruction to operate the bypass register. This command reduces the shift path to speed up serial data transfer of the other LSIs on the printed-circuit board. While executing this command, the test circuit has no effect on the system circuits. The code of the BYPASS command is 4'b1111.

(2) SAMPLE/PRELOAD

The SAMPLE/PRELOAD command inputs a value into the boundary scan register from the internal circuit of this LSI, and outputs data from or loads data to the scan path. When this command is executed, a value input to the input pin of this LSI is transferred to the internal circuit and output as it is to the outside through the output pin. While executing this command, the test circuit has no effect on the system circuits. The code of the SAMPLE/PRELOAD command is 4’b0001.

In sampling, a snapshot of the value transferred from the input pin to the internal circuit and from the internal circuit of the output pin is captured into the boundary scan register and read out from the scan path. Capturing a snapshot synchronizes with the rising edge of TCK in the Capture-DR state. Capturing does not interfere with the normal operation of this LSI.

In preloading, in advance of the EXTEST command, an initial value is set in the parallel output latch of the boundary scan register from the scan path. Without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence (transfer to the output latch) is completed (the EXTEST command consistently outputs the parallel output latch to the output pin).

(3) EXTEST

The EXTEST command conducts a test on the external circuits when mounting this LSI on the printed-circuit board. When this command is executed, the output pin outputs the test data (data set by the SAMPLE/PRELOAD command) from the boundary scan register to the printed-circuit board, and the input pin takes the test result from the printed-circuit board to the boundary scan register.

Since the test circuit controls the pins when this command is executed, the on-chip modules including the CPU of this LSI are set in the reset state.

Therefore, to change operation mode from EXTEST to another (mode in which the chip operates normally), set $\overline{\text{ASEMD0}}$, FWE, MD1, and MD0 to the desired operating mode, drive $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ low at the same time for a predetermined period of time, and input the clock signal to EXTAL. The code of the EXTEST command is 4'b0000.

(4) CLAMP

When the CLAMP command is set, the output pin outputs the value in the boundary scan register preset with the SAMPLE/PRELOAD command. Since the test circuit controls the pins when this command is executed, the on-chip modules including the CPU of this LSI are set in the reset state.

Therefore, to change operation mode from CLAMP to another (mode in which the chip operates normally), set $\overline{\text{ASEMD0}}$, FWE, MD1, and MD0 to the desired operating mode, drive $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ low at the same time for a predetermined period of time, and input the clock signal to EXTAL. The code of the CLAMP command is 4'b0010.

(5) HIGHZ

When the HIGHZ command is set, all the output pins for the boundary scan function except the $\overline{\text{WDTOVF}}$ pin are set to the high impedance state.

Since the test circuit controls the pins when this command is executed, the on-chip modules including the CPU of this LSI are set in the reset state.

Therefore, to change operation mode from HIGHZ to another (mode in which the chip operates normally), set $\overline{\text{ASEMD0}}$, FWE, MD1, and MD0 to the desired operating mode, drive $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ low at the same time for a predetermined period of time, and input the clock signal to EXTAL. The code of the HIGHZ command is 4'b0011.

(6) IDCODE

When the IDCODE is set, IDCODE (H'08083447) of this LSI is output from LSB to the TDO pin if the TAP controller is in the Shift-DR state. While executing this command, the test circuit has no effect on the system circuits. The code of the IDCODE command is 4'b0100.

31.7 Usage Notes

1. The following pins are not subject to the boundary scan function.
   — Clock-related pins (EXTAL, XTAL, USBEXTAL, and USBXTAL)
   — USB-related pins (USD+ and USD-)
   — System-related pin ($\overline{\text{RES}}$)
   — H-UDI-related pins ($\overline{\text{TRST}}$, TMS, TCK, TDI, TDO, and $\overline{\text{ASEMD0}}$)
2. The PBI2 and PBI3 pins are provided with input boundary scan registers, but not with output (open drain output) boundary scan registers.
3. Even if a transition to HIGHZ is made, the $\overline{\text{WDTOVF}}$ pin is driven to the high level, but not at high impedance.
4. The maximum frequency of TCK is 25 MHz.
5. When the power is turned on, input a low level to $\overline{\text{TRST}}$ at the same time with $\overline{\text{RES}}$ for a predetermined period of time and input the clock signal to EXTAL. Since TCK, TMS, and TDI are pulled up within the LSI, a current constantly flows if there is a difference in the electrical potentials between the input voltage of the pin and power supply voltage when the boundary scan function is not in use. Be careful especially when in the standby state.
6. A transition to EXTEST, CLAMP, or HIGHZ resets the internal modules including the CPU of this LSI. To make a transition to another mode from one of these, set $\overline{\text{ASEMD0}}$, FWE, MD1, and MD0 to the desired operation mode, input a low level to $\overline{\text{RES}}$ and $\overline{\text{TRST}}$ at the same time for a predetermined period of time, and input the clock signal to EXTAL.
7. An H-UDI command, once set, will not be modified as long as another command is not set again. If the same command is to be set continuously, the command must be set after a command (BYPASS, etc.) that does not affect chip operations is once set.
8. The H-UDI is used for emulator connection and therefore the boundary scan and H-UDI functions described in this section cannot be used when an emulator is used.
9. Fix the TMS pin to the high level for 200 ns after negating the signal on the $\overline{\text{TRST}}$ pin.
10. When the $\overline{\text{WDTOVF}}$ pin is being held at the high level due to a boundary scan, proceed after negating the signal on the $\overline{\text{RES}}$ pin.

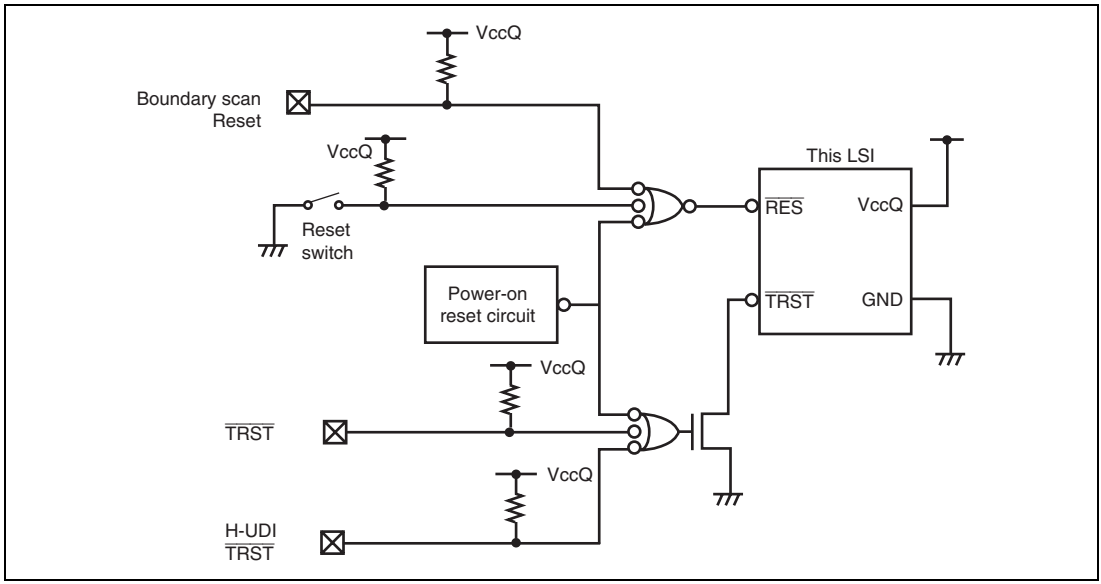


Figure 31.5 Peripheral Circuit Example of $\overline{\text{RES}}$ and $\overline{\text{TRST}}$

Section 32 List of Registers

This section gives information on the on-chip I/O registers of this LSI in the following structures.

1. Register Addresses (by functional module, in order of the corresponding section numbers)

- Registers are described by functional module, in order of the corresponding section numbers.
- Access to reserved addresses which are not described in this register address list is prohibited.
- When registers consist of 16 or 32 bits, the addresses of the MSBs are given when big-endian mode is selected.

2. Register Bits

- Bit configurations of the registers are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- Reserved bits are indicated by — in the bit name.
- No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (by functional module, in order of the corresponding section numbers).
- For the initial state of each bit, refer to the description of the register in the corresponding section.
- The register states described are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

4. Notes when Writing to the On-Chip Peripheral Modules

To access an on-chip module register, two or more peripheral module clock (Pf) cycles are required. Care must be taken in system design. When the CPU writes data to the internal peripheral registers, the CPU performs the succeeding instructions without waiting for the completion of writing to registers. For example, a case is described here in which the system is transferring to the software standby mode for power savings. To make this transition, the SLEEP instruction must be performed after setting the STBY bit in the STBCR register to 1. However a dummy read of the STBCR register is required before executing the SLEEP instruction. If a dummy read is omitted, the CPU executes the SLEEP instruction before the STBY bit is set to 1, thus the system enters sleep mode not software standby mode. A dummy read of the STBCR register is indispensable to complete writing to the STBY bit. To reflect the change by internal peripheral registers while performing the succeeding instructions, execute a dummy read of registers to which write instruction is given and then perform the succeeding instructions.

32.1 Register Addresses (by Functional Module, in Order of the Corresponding Section Numbers)

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CPG	Frequency control register	FRQCR	16	H'FFFE0010	16
	MTU2S clock frequency control register	MCLKCR	8	H'FFFE0410	8
	AD clock frequency control register	ACLKCR	8	H'FFFE0414	8
	Oscillation stop detection control register	OSCCR	8	H'FFFE001C	8
INTC	Interrupt control register 0	ICR0	16	H'FFFE0800	16, 32
	Interrupt control register 1	ICR1	16	H'FFFE0802	16
	IRQ interrupt request register	IRQRR	16	H'FFFE0806	16
	Bank control register	IBCR	16	H'FFFE080C	16, 32
	Bank number register	IBNR	16	H'FFFE080E	16
	Interrupt priority register 01	IPR01	16	H'FFFE0818	16, 32
	Interrupt priority register 02	IPR02	16	H'FFFE081A	16
	Interrupt priority register 05	IPR05	16	H'FFFE0820	16
	Interrupt priority register 06	IPR06	16	H'FFFE0C00	16, 32
	Interrupt priority register 07	IPR07	16	H'FFFE0C02	16
	Interrupt priority register 08	IPR08	16	H'FFFE0C04	16, 32
	Interrupt priority register 09	IPR09	16	H'FFFE0C06	16
	Interrupt priority register 10	IPR10	16	H'FFFE0C08	16, 32
	Interrupt priority register 11	IPR11	16	H'FFFE0C0A	16
	Interrupt priority register 12	IPR12	16	H'FFFE0C0C	16, 32
	Interrupt priority register 13	IPR13	16	H'FFFE0C0E	16
	Interrupt priority register 14	IPR14	16	H'FFFE0C10	16, 32
	Interrupt priority register 15	IPR15	16	H'FFFE0C12	16
	Interrupt priority register 16	IPR16	16	H'FFFE0C14	16, 32
	Interrupt priority register 17	IPR17	16	H'FFFE0C16	16
	Interrupt priority register 18	IPR18	16	H'FFFE0C18	16, 32
	Interrupt priority register 19	IPR19	16	H'FFFE0C1A	16
	USB-DTC transfer interrupt request register	USDTENDRR	16	H'FFFE0C50	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
UBC	Break address register_0	BAR_0	32	H'FFFC0400	32
	Break address mask register_0	BAMR_0	32	H'FFFC0404	32
	Break bus cycle register_0	BBR_0	16	H'FFFC04A0	16
	Break address register_1	BAR_1	32	H'FFFC0410	32
	Break address mask register_1	BAMR_1	32	H'FFFC0414	32
	Break bus cycle register_1	BBR_1	16	H'FFFC04B0	16
	Break address register_2	BAR_2	32	H'FFFC0420	32
	Break address mask register_2	BAMR_2	32	H'FFFC0424	32
	Break bus cycle register_2	BBR_2	16	H'FFFC04A4	16
	Break address register_3	BAR_3	32	H'FFFC0430	32
	Break address mask register_3	BAMR_3	32	H'FFFC0434	32
	Break bus cycle register_3	BBR_3	16	H'FFFC04B4	16
	Break control register	BRCR	32	H'FFFC04C0	32
DTC	DTC enable register A	DTCERA	16	H'FFFE6000	8, 16
	DTC enable register B	DTCERB	16	H'FFFE6002	8, 16
	DTC enable register C	DTCERC	16	H'FFFE6004	8, 16
	DTC enable register D	DTCERD	16	H'FFFE6006	8, 16
	DTC enable register E	DTCERE	16	H'FFFE6008	8, 16
	DTC control register	DTCCR	8	H'FFFE6010	8
	DTC vector base register	DTCVBR	32	H'FFFE6014	8, 16, 32
BSC	Common control register	CMNCR	32	H'FFFC0000	32
	CS0 space bus control register	CS0BCR	32	H'FFFC0004	32
	CS1 space bus control register	CS1BCR	32	H'FFFC0008	32
	CS2 space bus control register	CS2BCR	32	H'FFFC000C	32
	CS3 space bus control register	CS3BCR	32	H'FFFC0010	32
	CS4 space bus control register	CS4BCR	32	H'FFFC0014	32
	CS5 space bus control register	CS5BCR	32	H'FFFC0018	32
	CS6 space bus control register	CS6BCR	32	H'FFFC001C	32
	CS7 space bus control register	CS7BCR	32	H'FFFC0020	32
	CS0 space wait control register	CS0WCR	32	H'FFFC0028	32
	CS1 space wait control register	CS1WCR	32	H'FFFC002C	32
	CS2 space wait control register	CS2WCR	32	H'FFFC0030	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
BSC	CS3 space wait control register	CS3WCR	32	H'FFFC0034	32
	CS4 space wait control register	CS4WCR	32	H'FFFC0038	32
	CS5 space wait control register	CS5WCR	32	H'FFFC003C	32
	CS6 space wait control register	CS6WCR	32	H'FFFC0040	32
	CS7 space wait control register	CS7WCR	32	H'FFFC0044	32
	SDRAM control register	SDCR	32	H'FFFC004C	32
	Refresh timer control/status register	RTCSR	32	H'FFFC0050	32
	Refresh timer counter	RTCNT	32	H'FFFC0054	32
	Refresh time constant register	RTCOR	32	H'FFFC0058	32
	Bus function extending register	BSCEHR	16	H'FFFE3C1A	16
DMAC	DMA source address register_0	SAR_0	32	H'FFFE1000	16, 32
	DMA destination address register_0	DAR_0	32	H'FFFE1004	16, 32
	DMA transfer count register_0	DMATCR_0	32	H'FFFE1008	16, 32
	DMA channel control register_0	CHCR_0	32	H'FFFE100C	8, 16, 32
	DMA reload source address register_0	RSAR_0	32	H'FFFE1100	16, 32
	DMA reload destination address register_0	RDAR_0	32	H'FFFE1104	16, 32
	DMA reload transfer count register_0	RDMATCR_0	32	H'FFFE1108	16, 32
	DMA source address register_1	SAR_1	32	H'FFFE1010	16, 32
	DMA destination address register_1	DAR_1	32	H'FFFE1014	16, 32
	DMA transfer count register_1	DMATCR_1	32	H'FFFE1018	16, 32
	DMA channel control register_1	CHCR_1	32	H'FFFE101C	8, 16, 32
	DMA reload source address register_1	RSAR_1	32	H'FFFE1110	16, 32
	DMA reload destination address register_1	RDAR_1	32	H'FFFE1114	16, 32
	DMA reload transfer count register_1	RDMATCR_1	32	H'FFFE1118	16, 32
	DMA source address register_2	SAR_2	32	H'FFFE1020	16, 32
	DMA destination address register_2	DAR_2	32	H'FFFE1024	16, 32
	DMA transfer count register_2	DMATCR_2	32	H'FFFE1028	16, 32
	DMA channel control register_2	CHCR_2	32	H'FFFE102C	8, 16, 32
	DMA reload source address register_2	RSAR_2	32	H'FFFE1120	16, 32
	DMA reload destination address register_2	RDAR_2	32	H'FFFE1124	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA reload transfer count register_2	RDMATCR_2	32	H'FFFE1128	16, 32
	DMA source address register_3	SAR_3	32	H'FFFE1030	16, 32
	DMA destination address register_3	DAR_3	32	H'FFFE1034	16, 32
	DMA transfer count register_3	DMATCR_3	32	H'FFFE1038	16, 32
	DMA channel control register_3	CHCR_3	32	H'FFFE103C	8, 16, 32
	DMA reload source address register_3	RSAR_3	32	H'FFFE1130	16, 32
	DMA reload destination address register_3	RDAR_3	32	H'FFFE1134	16, 32
	DMA reload transfer count register_3	RDMATCR_3	32	H'FFFE1138	16, 32
	DMA source address register_4	SAR_4	32	H'FFFE1040	16, 32
	DMA destination address register_4	DAR_4	32	H'FFFE1044	16, 32
	DMA transfer count register_4	DMATCR_4	32	H'FFFE1048	16, 32
	DMA channel control register_4	CHCR_4	32	H'FFFE104C	8, 16, 32
	DMA reload source address register_4	RSAR_4	32	H'FFFE1140	16, 32
	DMA reload destination address register_4	RDAR_4	32	H'FFFE1144	16, 32
	DMA reload transfer count register_4	RDMATCR_4	32	H'FFFE1148	16, 32
	DMA source address register_5	SAR_5	32	H'FFFE1050	16, 32
	DMA destination address register_5	DAR_5	32	H'FFFE1054	16, 32
	DMA transfer count register_5	DMATCR_5	32	H'FFFE1058	16, 32
	DMA channel control register_5	CHCR_5	32	H'FFFE105C	8, 16, 32
	DMA reload source address register_5	RSAR_5	32	H'FFFE1150	16, 32
	DMA reload destination address register_5	RDAR_5	32	H'FFFE1154	16, 32
	DMA reload transfer count register_5	RDMATCR_5	32	H'FFFE1158	16, 32
	DMA source address register_6	SAR_6	32	H'FFFE1060	16, 32
	DMA destination address register_6	DAR_6	32	H'FFFE1064	16, 32
	DMA transfer count register_6	DMATCR_6	32	H'FFFE1068	16, 32
	DMA channel control register_6	CHCR_6	32	H'FFFE106C	8, 16, 32
	DMA reload source address register_6	RSAR_6	32	H'FFFE1160	16, 32
	DMA reload destination address register_6	RDAR_6	32	H'FFFE1164	16, 32
	DMA reload transfer count register_6	RDMATCR_6	32	H'FFFE1168	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	DMA source address register_7	SAR_7	32	H'FFFE1070	16, 32
	DMA destination address register_7	DAR_7	32	H'FFFE1074	16, 32
	DMA transfer count register_7	DMATCR_7	32	H'FFFE1078	16, 32
	DMA channel control register_7	CHCR_7	32	H'FFFE107C	8, 16, 32
	DMA reload source address register_7	RSAR_7	32	H'FFFE1170	16, 32
	DMA reload destination address register_7	RDAR_7	32	H'FFFE1174	16, 32
	DMA reload transfer count register_7	RDMATCR_7	32	H'FFFE1178	16, 32
	DMA operation register	DMAOR	16	H'FFFE1200	8, 16
	DMA extension resource selector 0	DMARS0	16	H'FFFE1300	16
	DMA extension resource selector 1	DMARS1	16	H'FFFE1304	16
	DMA extension resource selector 2	DMARS2	16	H'FFFE1308	16
	DMA extension resource selector 3	DMARS3	16	H'FFFE130C	16
MTU2	Timer control register_0	TCR_0	8	H'FFFE4300	8, 16, 32
	Timer mode register_0	TMDR_0	8	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	8	H'FFFE4302	8, 16
	Timer I/O control register L_0	TIORL_0	8	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	8	H'FFFE4304	8, 16, 32
	Timer status register_0	TSR_0	8	H'FFFE4305	8
	Timer counter_0	TCNT_0	16	H'FFFE4306	16
	Timer general register A_0	TGRA_0	16	H'FFFE4308	16, 32
	Timer general register B_0	TGRB_0	16	H'FFFE430A	16
	Timer general register C_0	TGRC_0	16	H'FFFE430C	16, 32
	Timer general register D_0	TGRD_0	16	H'FFFE430E	16
	Timer general register E_0	TGRE_0	16	H'FFFE4320	16, 32
	Timer general register F_0	TGRF_0	16	H'FFFE4322	16
	Timer interrupt enable register2_0	TIER2_0	8	H'FFFE4324	8, 16
	Timer status register2_0	TSR2_0	8	H'FFFE4325	8
	Timer buffer operation transfer mode register_0	TBTM_0	8	H'FFFE4326	8
	Timer control register_1	TCR_1	8	H'FFFE4380	8, 16
	Timer mode register_1	TMDR_1	8	H'FFFE4381	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer I/O control register_1	TIOR_1	8	H'FFFE4382	8
	Timer interrupt enable register_1	TIER_1	8	H'FFFE4384	8, 16, 32
	Timer status register_1	TSR_1	8	H'FFFE4385	8
	Timer counter_1	TCNT_1	16	H'FFFE4386	16
	Timer general register A_1	TGRA_1	16	H'FFFE4388	16, 32
	Timer general register B_1	TGRB_1	16	H'FFFE438A	16
	Timer input capture control register	TICCR	8	H'FFFE4390	8
	Timer control register_2	TCR_2	8	H'FFFE4000	8, 16
	Timer mode register_2	TMDR_2	8	H'FFFE4001	8
	Timer I/O control register_2	TIOR_2	8	H'FFFE4002	8
	Timer interrupt enable register_2	TIER_2	8	H'FFFE4004	8, 16, 32
	Timer status register_2	TSR_2	8	H'FFFE4005	8
	Timer counter_2	TCNT_2	16	H'FFFE4006	16
	Timer general register A_2	TGRA_2	16	H'FFFE4008	16, 32
	Timer general register B_2	TGRB_2	16	H'FFFE400A	16
	Timer control register_3	TCR_3	8	H'FFFE4200	8, 16, 32
	Timer mode register_3	TMDR_3	8	H'FFFE4202	8, 16
	Timer I/O control register H_3	TIORH_3	8	H'FFFE4204	8, 16, 32
	Timer I/O control register L_3	TIORL_3	8	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	8	H'FFFE4208	8, 16
	Timer status register_3	TSR_3	8	H'FFFE422C	8, 16
	Timer counter_3	TCNT_3	16	H'FFFE4210	16, 32
	Timer general register A_3	TGRA_3	16	H'FFFE4218	16, 32
	Timer general register B_3	TGRB_3	16	H'FFFE421A	16
	Timer general register C_3	TGRC_3	16	H'FFFE4224	16, 32
	Timer general register D_3	TGRD_3	16	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238	8, 16
	Timer control register_4	TCR_4	8	H'FFFE4201	8
	Timer mode register_4	TMDR_4	8	H'FFFE4203	8
	Timer I/O control register H_4	TIORH_4	8	H'FFFE4206	8, 16
	Timer I/O control register L_4	TIORL_4	8	H'FFFE4207	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer interrupt enable register_4	TIER_4	8	H'FFFE4209	8
	Timer status register_4	TSR_4	8	H'FFFE422D	8
	Timer counter_4	TCNT_4	16	H'FFFE4212	16
	Timer general register A_4	TGRA_4	16	H'FFFE421C	16, 32
	Timer general register B_4	TGRB_4	16	H'FFFE421E	16
	Timer general register C_4	TGRC_4	16	H'FFFE4228	16, 32
	Timer general register D_4	TGRD_4	16	H'FFFE422A	16
	Timer buffer operation transfer mode register_4	TBTM_4	8	H'FFFE4239	8
	Timer A/D converter start request control register	TADCR	16	H'FFFE4240	16
	Timer A/D converter start request cycle set register A	TADCORA_4	16	H'FFFE4244	16, 32
	Timer A/D converter start request cycle set register B_4	TADCORB_4	16	H'FFFE4246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	16	H'FFFE4248	16, 32
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	16	H'FFFE424A	16
	Timer control register U_5	TCRU_5	8	H'FFFE4084	8
	Timer control register V_5	TCRV_5	8	H'FFFE4094	8
	Timer control register W_5	TCRW_5	8	H'FFFE40A4	8
	Timer I/O control register U_5	TIORU_5	8	H'FFFE4086	8
	Timer I/O control register V_5	TIORV_5	8	H'FFFE4096	8
	Timer I/O control register W_5	TIORW_5	8	H'FFFE40A6	8
	Timer interrupt enable register_5	TIER_5	8	H'FFFE40B2	8
	Timer status register_5	TSR_5	8	H'FFFE40B0	8
	Timer start register_5	TSTR_5	8	H'FFFE40B4	8
	Timer counter U_5	TCNTU_5	16	H'FFFE4080	16, 32
	Timer counter V_5	TCNTV_5	16	H'FFFE4090	16, 32
	Timer counter W_5	TCNTW_5	16	H'FFFE40A0	16, 32
	Timer general register U_5	TGRU_5	16	H'FFFE4082	16
	Timer general register V_5	TGRV_5	16	H'FFFE4092	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer general register W_5	TGRW_5	16	H'FFFE40A2	16
	Timer compare match clear register	TCNTCMPCLR	8	H'FFFE40B6	8
	Timer start register	TSTR	8	H'FFFE4280	8, 16
	Timer synchronous register	TSYR	8	H'FFFE4281	8
	Timer counter synchronous start register	TCSYSTR	8	H'FFFE4282	8
	Timer read/write enable register	TRWER	8	H'FFFE4284	8
	Timer output master enable register	TOER	8	H'FFFE420A	8
	Timer output control register 1	TOCR1	8	H'FFFE420E	8, 16
	Timer output control register 2	TOCR2	8	H'FFFE420F	8
	Timer gate control register	TGCR	8	H'FFFE420D	8
	Timer cycle control register	TCDR	16	H'FFFE4214	16, 32
	Timer dead time data register	TDDR	16	H'FFFE4216	16
	Timer subcounter	TCNTS	16	H'FFFE4220	16, 32
	Timer cycle buffer register	TCBR	16	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	8	H'FFFE4230	8, 16
	Timer interrupt skipping counter	TITCNT	8	H'FFFE4231	8
	Timer buffer transfer set register	TBTER	8	H'FFFE4232	8
	Timer dead time enable register	TDER	8	H'FFFE4234	8
	Timer waveform control register	TWCR	8	H'FFFE4260	8
	Timer output level buffer register	TOLBR	8	H'FFFE4236	8
MTU2S	Timer control register_3S	TCR_3S	8	H'FFFE4A00	8, 16, 32
	Timer mode register_3S	TMDR_3S	8	H'FFFE4A02	8, 16
	Timer I/O control register H_3S	TIORH_3S	8	H'FFFE4A04	8, 16, 32
	Timer I/O control register L_3S	TIORL_3S	8	H'FFFE4A05	8
	Timer interrupt enable register_3S	TIER_3S	8	H'FFFE4A08	8, 16
	Timer status register_3S	TSR_3S	8	H'FFFE4A2C	8, 16
	Timer counter_3S	TCNT_3S	16	H'FFFE4A10	16, 32
	Timer general register A_3S	TGRA_3S	16	H'FFFE4A18	16, 32
	Timer general register B_3S	TGRB_3S	16	H'FFFE4A1A	16
	Timer general register C_3S	TGRC_3S	16	H'FFFE4A24	16, 32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2S	Timer general register D_3S	TGRD_3S	16	H'FFFE4A26	16
	Timer buffer operation transfer mode register_3S	TBTM_3S	8	H'FFFE4A38	8, 16
	Timer control register_4S	TCR_4S	8	H'FFFE4A01	8
	Timer mode register_4S	TMDR_4S	8	H'FFFE4A03	8
	Timer I/O control register H_4S	TIORH_4S	8	H'FFFE4A06	8, 16
	Timer I/O control register L_4S	TIORL_4S	8	H'FFFE4A07	8
	Timer interrupt enable register_4S	TIER_4S	8	H'FFFE4A09	8
	Timer status register_4S	TSR_4S	8	H'FFFE4A2D	8
	Timer counter_4S	TCNT_4S	16	H'FFFE4A12	16
	Timer general register A_4S	TGRA_4S	16	H'FFFE4A1C	16, 32
	Timer general register B_4S	TGRB_4S	16	H'FFFE4A1E	16
	Timer general register C_4S	TGRC_4S	16	H'FFFE4A28	16, 32
	Timer general register D_4S	TGRD_4S	16	H'FFFE4A2A	16
	Timer buffer operation transfer mode register_4S	TBTM_4S	8	H'FFFE4A39	8
	Timer A/D converter start request control register S	TADCRS	16	H'FFFE4A40	16
	Timer A/D converter start request cycle set register A_4S	TADCORA_4S	16	H'FFFE4A44	16, 32
	Timer A/D converter start request cycle set register B_4S	TADCORB_4S	16	H'FFFE4A46	16
	Timer A/D converter start request cycle set buffer register A_4S	TADCOBRA_4S	16	H'FFFE4A48	16, 32
	Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	16	H'FFFE4A4A	16
	Timer control register U_5S	TCRU_5S	8	H'FFFE4884	8
	Timer control register V_5S	TCRV_5S	8	H'FFFE4894	8
	Timer control register W_5S	TCRW_5S	8	H'FFFE48A4	8
	Timer I/O control register U_5S	TIORU_5S	8	H'FFFE4886	8
	Timer I/O control register V_5S	TIORV_5S	8	H'FFFE4896	8
	Timer I/O control register W_5S	TIORW_5S	8	H'FFFE48A6	8
	Timer interrupt enable register_5S	TIER_5S	8	H'FFFE48B2	8
	Timer status register_5S	TSR_5S	8	H'FFFE48B0	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2S	Timer start register_5S	TSTR_5S	8	H'FFFE48B4	8
	Timer counter U_5S	TCNTU_5S	16	H'FFFE4880	16, 32
	Timer counter V_5S	TCNTV_5S	16	H'FFFE4890	16, 32
	Timer counter W_5S	TCNTW_5S	16	H'FFFE48A0	16, 32
	Timer general register U_5S	TGRU_5S	16	H'FFFE4882	16
	Timer general register V_5S	TGRV_5S	16	H'FFFE4892	16
	Timer general register W_5S	TGRW_5S	16	H'FFFE48A2	16
	Timer compare match clear register S	TCNTCMPCLRS	8	H'FFFE48B6	8
	Timer start register S	TSTRS	8	H'FFFE4A80	8, 16
	Timer synchronous register S	TSYRS	8	H'FFFE4A81	8
	Timer read/write enable register S	TRWERS	8	H'FFFE4A84	8
	Timer output master enable register S	TOERS	8	H'FFFE4A0A	8
	Timer output control register 1S	TOCR1S	8	H'FFFE4A0E	8, 16
	Timer output control register 2S	TOCR2S	8	H'FFFE4A0F	8
	Timer gate control register S	TGCRS	8	H'FFFE4A0D	8
	Timer cycle data register S	TCDRS	16	H'FFFE4A14	16, 32
	Timer dead time data register S	TDDRS	16	H'FFFE4A16	16
	Timer subcounter S	TCNTSS	16	H'FFFE4A20	16, 32
	Timer cycle buffer register S	TCBRS	16	H'FFFE4A22	16
	Timer interrupt skipping set register S	TITCRS	8	H'FFFE4A30	8, 16
	Timer interrupt skipping counter S	TITCNTS	8	H'FFFE4A31	8
	Timer buffer transfer set register S	TBTERS	8	H'FFFE4A32	8
	Timer dead time enable register S	TDERS	8	H'FFFE4A34	8
	Timer synchronous clear register S	TSYCRS	8	H'FFFE4A50	8
	Timer waveform control register S	TWCRS	8	H'FFFE4A60	8
	Timer output level buffer register S	TOLBRS	8	H'FFFE4A36	8
POE2	Input level control/status register 1	ICSR1	16	H'FFFE5000	16
	Output level control/status register 1	OCSR1	16	H'FFFE5002	16
	Input level control/status register 2	ICSR2	16	H'FFFE5004	16
	Output level control/status register 2	OCSR2	16	H'FFFE5006	16
	Input level control/status register 3	ICSR3	16	H'FFFE5008	16
	Software port output enable register	SPOER	8	H'FFFE500A	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
POE2	Port output enable control register 1	POECR1	8	H'FFFE500B	8
	Port output enable control register 2	POECR2	16	H'FFFE500C	16
CMT	Compare match timer start register	CMSTR	16	H'FFFEC000	16
	Compare match timer control/ status register_0	CMCSR_0	16	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	16	H'FFFEC004	16
	Compare match constant register_0	CMCOR_0	16	H'FFFEC006	16
	Compare match timer control/ status register_1	CMCSR_1	16	H'FFFEC008	16
	Compare match counter_1	CMCNT_1	16	H'FFFEC00A	16
	Compare match constant register_1	CMCOR_1	16	H'FFFEC00C	16
WDT	Watchdog timer control/status register	WTCSR	16	H'FFFE0000	*1
	Watchdog timer counter	WTCNT	16	H'FFFE0002	*1
	Watchdog reset control/status register	WRCSR	16	H'FFFE0004	*1
SCI (channel 0)	Serial mode register_0	SCSMR_0	8	H'FFFF8000	8
	Bit rate register_0	SCBRR_0	8	H'FFFF8002	8
	Serial control register_0	SCSCR_0	8	H'FFFF8004	8
	Transmit data register_0	SCTDR_0	8	H'FFFF8006	8
	Serial status register_0	SCSSR_0	8	H'FFFF8008	8
	Receive data register_0	SCRDR_0	8	H'FFFF800A	8
	Serial direction control register_0	SCSDCR_0	8	H'FFFF800C	8
	Serial port register_0	SCSPTR_0	8	H'FFFF800E	8
SCI (channel 1)	Serial mode register_1	SCSMR_1	8	H'FFFF8800	8
	Bit rate register_1	SCBRR_1	8	H'FFFF8802	8
	Serial control register_1	SCSCR_1	8	H'FFFF8804	8
	Transmit data register_1	SCTDR_1	8	H'FFFF8806	8
	Serial status register_1	SCSSR_1	8	H'FFFF8808	8
	Receive data register_1	SCRDR_1	8	H'FFFF880A	8
	Serial direction control register_1	SCSDCR_1	8	H'FFFF880C	8
	Serial port register_1	SCSPTR_1	8	H'FFFF880E	8
SCI (channel 2)	Serial mode register_2	SCSMR_2	8	H'FFFF9000	8
	Bit rate register_2	SCBRR_2	8	H'FFFF9002	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCI (channel 2)	Serial control register_2	SCSCR_2	8	H'FFFF9004	8
	Transmit data register_2	SCTDR_2	8	H'FFFF9006	8
	Serial status register_2	SCSSR_2	8	H'FFFF9008	8
	Receive data register_2	SCRDR_2	8	H'FFFF900A	8
	Serial direction control register_2	SCSDCR_2	8	H'FFFF900C	8
	Serial port register_2	SCSPTR_2	8	H'FFFF900E	8
SCI (channel 4)	Serial mode register_4	SCSMR_4	8	H'FFFFA000	8
	Bit rate register_4	SCBRR_4	8	H'FFFFA002	8
	Serial control register_4	SCSCR_4	8	H'FFFFA004	8
	Transmit data register_4	SCTDR_4	8	H'FFFFA006	8
	Serial status register_4	SCSSR_4	8	H'FFFFA008	8
	Receive data register_4	SCRDR_4	8	H'FFFFA00A	8
	Serial direction control register_4	SCSDCR_4	8	H'FFFFA00C	8
	Serial port register_4	SCSPTR_4	8	H'FFFFA00E	8
SCIF	Serial mode register_3	SCSMR_3	16	H'FFFE9800	16
	Bit rate register_3	SCBRR_3	8	H'FFFE9804	8
	Serial control register_3	SCSCR_3	16	H'FFFE9808	16
	Transmit FIFO data register_3	SCFTDR_3	8	H'FFFE980C	8
	Serial status register_3	SCFSR_3	16	H'FFFE9810	16
	Receive FIFO data register_3	SCFRDR_3	8	H'FFFE9814	8
	FIFO control register_3	SCFCR_3	16	H'FFFE9818	16
	FIFO data count register_3	SCFDR_3	16	H'FFFE981C	16
	Serial port register_3	SCSPTR_3	16	H'FFFE9820	16
	Line status register_3	SCLSR_3	16	H'FFFE9824	16
	Serial extended mode register_3	SCSEMR_3	8	H'FFFE9900	8
RSPI	RSPI control register	SPCR	8	H'FFFFB000	8, 16
	RSPI slave select polarity register	SSLP	8	H'FFFFB001	8
	RSPI pin control register	SPPCR	8	H'FFFFB002	8, 16
	RSPI status register	SPSR	8	H'FFFFB003	8
	RSPI data register	SPDR	32	H'FFFFB004	16, 32*[2]
	RSPI sequence control register	SPSCR	8	H'FFFFB008	8, 16
	RSPI sequence status register	SPSSR	8	H'FFFFB009	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
RSPI	RSPI bit rate register	SPBR	8	H'FFFFB00A	8, 16
	RSPI data control register	SPDCR	8	H'FFFFB00B	8
	RSPI clock delay register	SPCKD	8	H'FFFFB00C	8, 16
	RSPI slave select negation delay register	SSLND	8	H'FFFFB00D	8
	RSPI next-access delay register	SPND	8	H'FFFFB00E	8
	RSPI command register 0	SPCMD0	16	H'FFFFB010	16
	RSPI command register 1	SPCMD1	16	H'FFFFB012	16
	RSPI command register 2	SPCMD2	16	H'FFFFB014	16
	RSPI command register 3	SPCMD3	16	H'FFFFB016	16
IIC3	I^2C bus control register 1	ICCR1	8	H'FFFEE000	8
	I^2C bus control register 2	ICCR2	8	H'FFFEE001	8
	I^2C bus mode register	ICMR	8	H'FFFEE002	8
	I^2C bus interrupt enable register	ICIER	8	H'FFFEE003	8
	I^2C bus status register	ICSR	8	H'FFFEE004	8
	Slave address register	SAR	8	H'FFFEE005	8
	I^2C bus transmit data register	ICDRT	8	H'FFFEE006	8
	I^2C bus receive data register	ICDRR	8	H'FFFEE007	8
	NF2CYC register	NF2CYC	8	H'FFFEE008	8
ADC	A/D control register_0	ADCR_0	8	H'FFFFE800	8
	A/D status register_0	ADSR_0	8	H'FFFFE802	8
	A/D start trigger select register_0	ADSTRGR_0	8	H'FFFFE81C	8
	A/D analog input channel select register_0	ADANSR_0	8	H'FFFFE820	8
	A/D bypass control register_0	ADBYPSCR_0	8	H'FFFFE830	8
	A/D data register 0	ADDR0	16	H'FFFFE840	16
	A/D data register 1	ADDR1	16	H'FFFFE842	16
	A/D data register 2	ADDR2	16	H'FFFFE844	16
	A/D data register 3	ADDR3	16	H'FFFFE846	16
	A/D control register_1	ADCR_1	8	H'FFFFEC00	8
	A/D status register_1	ADSR_1	8	H'FFFFEC02	8
	A/D start trigger select register_1	ADSTRGR_1	8	H'FFFFEC1C	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC	A/D analog input channel select register_1	ADANSR_1	8	H'FFFFEC20	8
	A/D bypass control register_1	ADBYPSCR_1	8	H'FFFFEC30	8
	A/D data register 4	ADDR4	16	H'FFFFEC40	16
	A/D data register 5	ADDR5	16	H'FFFFEC42	16
	A/D data register 6	ADDR6	16	H'FFFFEC44	16
	A/D data register 7	ADDR7	16	H'FFFFEC46	16
RCAN-ET	Master control register	MCR	16	H'FFFFD000	16
	General status register	GSR	16	H'FFFFD002	16
	Bit configuration register 1	BCR1	16	H'FFFFD004	16
	Bit configuration register 0	BCR0	16	H'FFFFD006	16
	Interrupt request register	IRR	16	H'FFFFD008	16
	Interrupt mask register	IMR	16	H'FFFFD00A	16
	Error counter register	TEC/REC	16	H'FFFFD00C	16
	Transmit pending 1, 0	TXPR1, 0	32	H'FFFFD020	32
	Transmit cancel 0	TXCR0	16	H'FFFFD02A	16
	Transmit acknowledge 0	TXACK0	16	H'FFFFD032	16
	Abort acknowledge 0	ABACK0	16	H'FFFFD03A	16
	Data frame receive pending 0	RXPR0	16	H'FFFFD042	16
	Remote frame receive pending 0	RFPR0	16	H'FFFFD04A	16
	Mailbox interrupt mask register 0	MBIMR0	16	H'FFFFD052	16
	Unread message status register 0	UMSR0	16	H'FFFFD05A	16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[0].	CONTROL0H	—	16	H'FFFFD100	16, 32
		CONTROL0L	—	16	H'FFFFD102	16
		LAFMH	—	16	H'FFFFD104	16, 32
		LAFML	—	16	H'FFFFD106	16
		MSG_DATA[0]	—	8	H'FFFFD108	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD109	8
		MSG_DATA[2]	—	8	H'FFFFD10A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD10B	8
		MSG_DATA[4]	—	8	H'FFFFD10C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD10D	8
		MSG_DATA[6]	—	8	H'FFFFD10E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD10F	8
		CONTROL1H	—	8	H'FFFFD110	8, 16
		CONTROL1L	—	8	H'FFFFD111	8
	MB[1].	CONTROL0H	—	16	H'FFFFD120	16, 32
		CONTROL0L	—	16	H'FFFFD122	16
		LAFMH	—	16	H'FFFFD124	16, 32
		LAFML	—	16	H'FFFFD126	16
		MSG_DATA[0]	—	8	H'FFFFD128	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD129	8
		MSG_DATA[2]	—	8	H'FFFFD12A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD12B	8
		MSG_DATA[4]	—	8	H'FFFFD12C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD12D	8
		MSG_DATA[6]	—	8	H'FFFFD12E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD12F	8
		CONTROL1H	—	8	H'FFFFD130	8, 16
		CONTROL1L	—	8	H'FFFFD131	8
	MB[2].	CONTROL0H	—	16	H'FFFFD140	16, 32
		CONTROL0L	—	16	H'FFFFD142	16
		LAFMH	—	16	H'FFFFD144	16, 32
		LAFML	—	16	H'FFFFD146	16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[2].	MSG_DATA[0]	—	8	H'FFFFD148	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD149	8
		MSG_DATA[2]	—	8	H'FFFFD14A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD14B	8
		MSG_DATA[4]	—	8	H'FFFFD14C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD14D	8
		MSG_DATA[6]	—	8	H'FFFFD14E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD14F	8
		CONTROL1H	—	8	H'FFFFD150	8, 16
		CONTROL1L	—	8	H'FFFFD151	8
	MB[3].	CONTROL0H	—	16	H'FFFFD160	16, 32
		CONTROL0L	—	16	H'FFFFD162	16
		LAFMH	—	16	H'FFFFD164	16, 32
		LAFML	—	16	H'FFFFD166	16
		MSG_DATA[0]	—	8	H'FFFFD168	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD169	8
		MSG_DATA[2]	—	8	H'FFFFD16A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD16B	8
		MSG_DATA[4]	—	8	H'FFFFD16C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD16D	8
		MSG_DATA[6]	—	8	H'FFFFD16E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD16F	8
		CONTROL1H	—	8	H'FFFFD170	8, 16
		CONTROL1L	—	8	H'FFFFD171	8
	MB[4].	CONTROL0H	—	16	H'FFFFD180	16, 32
		CONTROL0L	—	16	H'FFFFD182	16
		LAFMH	—	16	H'FFFFD184	16, 32
		LAFML	—	16	H'FFFFD186	16
		MSG_DATA[0]	—	8	H'FFFFD188	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD189	8
		MSG_DATA[2]	—	8	H'FFFFD18A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD18B	8

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[4].	MSG_DATA[4]	—	8	H'FFFFD18C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD18D	8
		MSG_DATA[6]	—	8	H'FFFFD18E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD18F	8
		CONTROL1H	—	8	H'FFFFD190	8, 16
		CONTROL1L	—	8	H'FFFFD191	8
	MB[5].	CONTROL0H	—	16	H'FFFFD1A0	16, 32
		CONTROL0L	—	16	H'FFFFD1A2	16
		LAFMH	—	16	H'FFFFD1A4	16, 32
		LAFML	—	16	H'FFFFD1A6	16
		MSG_DATA[0]	—	8	H'FFFFD1A8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD1A9	8
		MSG_DATA[2]	—	8	H'FFFFD1AA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD1AB	8
		MSG_DATA[4]	—	8	H'FFFFD1AC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD1AD	8
		MSG_DATA[6]	—	8	H'FFFFD1AE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD1AF	8
		CONTROL1H	—	8	H'FFFFD1B0	8, 16
		CONTROL1L	—	8	H'FFFFD1B1	8
	MB[6].	CONTROL0H	—	16	H'FFFFD1C0	16, 32
		CONTROL0L	—	16	H'FFFFD1C2	16
		LAFMH	—	16	H'FFFFD1C4	16, 32
		LAFML	—	16	H'FFFFD1C6	16
		MSG_DATA[0]	—	8	H'FFFFD1C8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD1C9	8
		MSG_DATA[2]	—	8	H'FFFFD1CA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD1CB	8
		MSG_DATA[4]	—	8	H'FFFFD1CC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD1CD	8
		MSG_DATA[6]	—	8	H'FFFFD1CE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD1CF	8

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[6].	CONTROL1H	—	8	H'FFFFD1D0	8, 16
		CONTROL1L	—	8	H'FFFFD1D1	8
	MB[7].	CONTROL0H	—	16	H'FFFFD1E0	16, 32
		CONTROL0L	—	16	H'FFFFD1E2	16
		LAFMH	—	16	H'FFFFD1E4	16, 32
		LAFML	—	16	H'FFFFD1E6	16
		MSG_DATA[0]	—	8	H'FFFFD1E8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD1E9	8
		MSG_DATA[2]	—	8	H'FFFFD1EA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD1EB	8
		MSG_DATA[4]	—	8	H'FFFFD1EC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD1ED	8
		MSG_DATA[6]	—	8	H'FFFFD1EE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD1EF	8
		CONTROL1H	—	8	H'FFFFD1F0	8, 16
		CONTROL1L	—	8	H'FFFFD1F1	8
	MB[8].	CONTROL0H	—	16	H'FFFFD200	16, 32
		CONTROL0L	—	16	H'FFFFD202	16
		LAFMH	—	16	H'FFFFD204	16, 32
		LAFML	—	16	H'FFFFD206	16
		MSG_DATA[0]	—	8	H'FFFFD208	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD209	8
		MSG_DATA[2]	—	8	H'FFFFD20A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD20B	8
		MSG_DATA[4]	—	8	H'FFFFD20C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD20D	8
		MSG_DATA[6]	—	8	H'FFFFD20E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD20F	8
		CONTROL1H	—	8	H'FFFFD210	8, 16
		CONTROL1L	—	8	H'FFFFD211	8
	MB[9].	CONTROL0H	—	16	H'FFFFD220	16, 32
		CONTROL0L	—	16	H'FFFFD222	16

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[9].	LAFMH	—	16	H'FFFFD224	16, 32
		LAFML	—	16	H'FFFFD226	16
		MSG_DATA[0]	—	8	H'FFFFD228	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD229	8
		MSG_DATA[2]	—	8	H'FFFFD22A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD22B	8
		MSG_DATA[4]	—	8	H'FFFFD22C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD22D	8
		MSG_DATA[6]	—	8	H'FFFFD22E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD22F	8
		CONTROL1H	—	8	H'FFFFD230	8, 16
		CONTROL1L	—	8	H'FFFFD231	8
	MB[10].	CONTROL0H	—	16	H'FFFFD240	16, 32
		CONTROL0L	—	16	H'FFFFD242	16
		LAFMH	—	16	H'FFFFD244	16, 32
		LAFML	—	16	H'FFFFD246	16
		MSG_DATA[0]	—	8	H'FFFFD248	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD249	8
		MSG_DATA[2]	—	8	H'FFFFD24A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD24B	8
		MSG_DATA[4]	—	8	H'FFFFD24C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD24D	8
		MSG_DATA[6]	—	8	H'FFFFD24E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD24F	8
		CONTROL1H	—	8	H'FFFFD250	8, 16
		CONTROL1L	—	8	H'FFFFD251	8
	MB[11].	CONTROL0H	—	16	H'FFFFD260	16, 32
		CONTROL0L	—	16	H'FFFFD262	16
		LAFMH	—	16	H'FFFFD264	16, 32
		LAFML	—	16	H'FFFFD266	16
		MSG_DATA[0]	—	8	H'FFFFD268	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD269	8

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[11].	MSG_DATA[2]	—	8	H'FFFFD26A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD26B	8
		MSG_DATA[4]	—	8	H'FFFFD26C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD26D	8
		MSG_DATA[6]	—	8	H'FFFFD26E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD26F	8
		CONTROL1H	—	8	H'FFFFD270	8, 16
		CONTROL1L	—	8	H'FFFFD271	8
	MB[12].	CONTROL0H	—	16	H'FFFFD280	16, 32
		CONTROL0L	—	16	H'FFFFD282	16
		LAFMH	—	16	H'FFFFD284	16, 32
		LAFML	—	16	H'FFFFD286	16
		MSG_DATA[0]	—	8	H'FFFFD288	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD289	8
		MSG_DATA[2]	—	8	H'FFFFD28A	8, 16
		MSG_DATA[3]	—	8	H'FFFFD28B	8
		MSG_DATA[4]	—	8	H'FFFFD28C	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD28D	8
		MSG_DATA[6]	—	8	H'FFFFD28E	8, 16
		MSG_DATA[7]	—	8	H'FFFFD28F	8
		CONTROL1H	—	8	H'FFFFD290	8, 16
		CONTROL1L	—	8	H'FFFFD291	8
	MB[13].	CONTROL0H	—	16	H'FFFFD2A0	16, 32
		CONTROL0L	—	16	H'FFFFD2A2	16
		LAFMH	—	16	H'FFFFD2A4	16, 32
		LAFML	—	16	H'FFFFD2A6	16
		MSG_DATA[0]	—	8	H'FFFFD2A8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD2A9	8
		MSG_DATA[2]	—	8	H'FFFFD2AA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD2AB	8
		MSG_DATA[4]	—	8	H'FFFFD2AC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD2AD	8

Module Name	Register Name		Abbreviation	Number of Bits	Address	Access Size
RCAN-ET	MB[13].	MSG_DATA[6]	—	8	H'FFFFD2AE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD2AF	8
		CONTROL1H	—	8	H'FFFFD2B0	8, 16
		CONTROL1L	—	8	H'FFFFD2B1	8
	MB[14].	CONTROL0H	—	16	H'FFFFD2C0	16, 32
		CONTROL0L	—	16	H'FFFFD2C2	16
		LAFMH	—	16	H'FFFFD2C4	16, 32
		LAFML	—	16	H'FFFFD2C6	16
		MSG_DATA[0]	—	8	H'FFFFD2C8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD2C9	8
		MSG_DATA[2]	—	8	H'FFFFD2CA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD2CB	8
		MSG_DATA[4]	—	8	H'FFFFD2CC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD2CD	8
		MSG_DATA[6]	—	8	H'FFFFD2CE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD2CF	8
		CONTROL1H	—	8	H'FFFFD2D0	8, 16
		CONTROL1L	—	8	H'FFFFD2D1	8
	MB[15].	CONTROL0H	—	16	H'FFFFD2E0	16, 32
		CONTROL0L	—	16	H'FFFFD2E2	16
		LAFMH	—	16	H'FFFFD2E4	16, 32
		LAFML	—	16	H'FFFFD2E6	16
		MSG_DATA[0]	—	8	H'FFFFD2E8	8, 16, 32
		MSG_DATA[1]	—	8	H'FFFFD2E9	8
		MSG_DATA[2]	—	8	H'FFFFD2EA	8, 16
		MSG_DATA[3]	—	8	H'FFFFD2EB	8
		MSG_DATA[4]	—	8	H'FFFFD2EC	8, 16, 32
		MSG_DATA[5]	—	8	H'FFFFD2ED	8
		MSG_DATA[6]	—	8	H'FFFFD2EE	8, 16
		MSG_DATA[7]	—	8	H'FFFFD2EF	8
		CONTROL1H	—	8	H'FFFFD2F0	8, 16
		CONTROL1L	—	8	H'FFFFD2F1	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port A IO register H	PAIORH	16	H'FFFE3804	8, 16, 32
	Port A IO register L	PAIORL	16	H'FFFE3806	8, 16
	Port A control register H2	PACRH2	16	H'FFFE380C	8, 16, 32
	Port A control register H1	PACRH1	16	H'FFFE380E	8, 16
	Port A control register L4	PACRL4	16	H'FFFE3810	8, 16, 32
	Port A control register L3	PACRL3	16	H'FFFE3812	8, 16
	Port A control register L2	PACRL2	16	H'FFFE3814	8, 16, 32
	Port A control register L1	PACRL1	16	H'FFFE3816	8, 16
	Port A pull-up MOS control register H	PAPCRH	16	H'FFFE3828	8, 16, 32
	Port A pull-up MOS control register L	PAPCRL	16	H'FFFE382A	8, 16
	Port B IO register L	PBIORL	16	H'FFFE3886	8, 16
	Port B control register L4	PBCRL4	16	H'FFFE3890	8, 16, 32
	Port B control register L3	PBCRL3	16	H'FFFE3892	8, 16
	Port B control register L2	PBCRL2	16	H'FFFE3894	8, 16, 32
	Port B control register L1	PBCRL1	16	H'FFFE3896	8, 16
	Port B pull-up MOS control register L	PBPCRL	16	H'FFFE38AA	8, 16
	Port C IO register L	PCIORL	16	H'FFFE3906	8, 16
	Port C control register L4	PCCRL4	16	H'FFFE3910	8, 16, 32
	Port C control register L3	PCCRL3	16	H'FFFE3912	8, 16
	Port C control register L2	PCCRL2	16	H'FFFE3914	8, 16, 32
	Port C control register L1	PCCRL1	16	H'FFFE3916	8, 16
	Port C pull-up MOS control register L	PCPCRL	16	H'FFFE392A	8, 16
	Port D IO register H	PDIORH	16	H'FFFE3984	8, 16, 32
	Port D IO register L	PDIORL	16	H'FFFE3986	8, 16
	Port D control register H4	PDCRH4	16	H'FFFE3988	8, 16, 32
	Port D control register H3	PDCRH3	16	H'FFFE398A	8, 16
	Port D control register H2	PDCRH2	16	H'FFFE398C	8, 16, 32
	Port D control register H1	PDCRH1	16	H'FFFE398E	8, 16
	Port D control register L4	PDCRL4	16	H'FFFE3990	8, 16, 32
	Port D control register L3	PDCRL3	16	H'FFFE3992	8, 16
	Port D control register L2	PDCRL2	16	H'FFFE3994	8, 16, 32
	Port D control register L1	PDCRL1	16	H'FFFE3996	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port D pull-up MOS control register H	PDPCRH	16	H'FFFE39A8	8, 16, 32
	Port D pull-up MOS control register L	PDPCRL	16	H'FFFE39AA	8, 16
	Port E IO register L	PEIORL	16	H'FFFE3A06	8, 16
	Port E control register L4	PECRL4	16	H'FFFE3A10	8, 16, 32
	Port E control register L3	PECRL3	16	H'FFFE3A12	8, 16
	Port E control register L2	PECRL2	16	H'FFFE3A14	8, 16, 32
	Port E control register L1	PECRL1	16	H'FFFE3A16	8, 16
	Large current port control register	HCPCR	16	H'FFFE3A20	8, 16, 32
	IRQOUT function control register	IFCR	16	H'FFFE3A22	8, 16
	Port E pull-up MOS control register L	PEPCRL	16	H'FFFE3A2A	8, 16
	DACK output timing control register	PDACKCR	16	H'FFFE3A2C	8, 16
I/O port	Port A data register H	PADRH	16	H'FFFE3800	8, 16, 32
	Port A data register L	PADRL	16	H'FFFE3802	8, 16
	Port A port register H	PAPRH	16	H'FFFE381C	8, 16, 32
	Port A port register L	PAPRL	16	H'FFFE381E	8, 16
	Port B data register L	PBDRL	16	H'FFFE3882	8, 16
	Port B port register L	PBPRL	16	H'FFFE389E	8, 16
	Port C data register L	PCDRL	16	H'FFFE3902	8, 16
	Port C port register L	PCPRL	16	H'FFFE391E	8, 16
	Port D data register H	PDDRH	16	H'FFFE3980	8, 16, 32
	Port D data register L	PDDRL	16	H'FFFE3982	8, 16
	Port D port register H	PDPRH	16	H'FFFE399C	8, 16, 32
	Port D port register L	PDPRL	16	H'FFFE399E	8, 16
	Port E data register L	PEDRL	16	H'FFFE3A02	8, 16
	Port E port register L	PEPRL	16	H'FFFE3A1E	8, 16
	Port F data register L	PFDRL	16	H'FFFE3A82	8, 16
USB	USB interrupt flag register 0	USBIFR0	8	H'FFFE7000	8
	USB interrupt flag register 1	USBIFR1	8	H'FFFE7001	8
	USB interrupt flag register 2	USBIFR2	8	H'FFFE7002	8
	USB interrupt flag register 3	USBIFR3	8	H'FFFE7003	8
	USB interrupt flag register 4	USBIFR4	8	H'FFFE7004	8
	USB interrupt enable register 0	USBIER0	8	H'FFFE7008	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	USB interrupt enable register 1	USBIER1	8	H'FFFE7009	8
	USB interrupt enable register 2	USBIER2	8	H'FFFE700A	8
	USB interrupt enable register 3	USBIER3	8	H'FFFE700B	8
	USB interrupt enable register 4	USBIER4	8	H'FFFE700C	8
	USB interrupt select register 0	USBISR0	8	H'FFFE7010	8
	USB interrupt select register 1	USBISR1	8	H'FFFE7011	8
	USB interrupt select register 2	USBISR2	8	H'FFFE7012	8
	USB interrupt select register 3	USBISR3	8	H'FFFE7013	8
	USB interrupt select register 4	USBISR4	8	H'FFFE7014	8
	USBEP0i data register	USBEPDR0i	8	H'FFFE7020	8, 16, 32
	USBEP0o data register	USBEPDR0o	8	H'FFFE7024	8, 16, 32
	USBEP0s data register	USBEPDR0s	8	H'FFFE7028	8, 16, 32
	USBEP1 data register	USBEPDR1	8	H'FFFE7030	8, 16, 32
	USBEP2 data register	USBEPDR2	8	H'FFFE7034	8, 16, 32
	USBEP3 data register	USBEPDR3	8	H'FFFE7038	8, 16, 32
	USBEP4 data register	USBEPDR4	8	H'FFFE7040	8, 16, 32
	USBEP5 data register	USBEPDR5	8	H'FFFE7044	8, 16, 32
	USBEP6 data register	USBEPDR6	8	H'FFFE7048	8, 16, 32
	USBEP7 data register	USBEPDR7	8	H'FFFE7050	8, 16, 32
	USBEP8 data register	USBEPDR8	8	H'FFFE7054	8, 16, 32
	USBEP9 data register	USBEPDR9	8	H'FFFE7058	8, 16, 32
	USBEP0o receive data size register	USBEPSZ0o	8	H'FFFE7080	8
	USBEP1 receive data size register	USBEPSZ1	8	H'FFFE7081	8
	USBEP4 receive data size register	USBEPSZ4	8	H'FFFE7082	8
	USBEP7 receive data size register	USBEPSZ7	8	H'FFFE7083	8
	USB data status register 0	USBDASTS0	8	H'FFFE7088	8
	USB data status register 1	USBDASTS1	8	H'FFFE7089	8
	USB data status register 2	USBDASTS2	8	H'FFFE708A	8
	USB data status register 3	USBDASTS3	8	H'FFFE708B	8
	USB trigger register 0	USBTRG0	8	H'FFFE7090	8
	USB trigger register 1	USBTRG1	8	H'FFFE7091	8
	USB trigger register 2	USBTRG2	8	H'FFFE7092	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
USB	USB trigger register 3	USBTRG3	8	H'FFFE7093	8
	USB FIFO clear register 0	USBFCLR0	8	H'FFFE7098	8
	USB FIFO clear register 1	USBFCLR1	8	H'FFFE7099	8
	USB FIFO clear register 2	USBFCLR2	8	H'FFFE709A	8
	USB FIFO clear register 3	USBFCLR3	8	H'FFFE709B	8
	USB endpoint stall register 0	USBEPSTL0	8	H'FFFE70A0	8
	USB endpoint stall register 1	USBEPSTL1	8	H'FFFE70A1	8
	USB endpoint stall register 2	USBEPSTL2	8	H'FFFE70A2	8
	USB endpoint stall register 3	USBEPSTL3	8	H'FFFE70A3	8
	USB stall status register 1	USBSTLSR1	8	H'FFFE70A9	8
	USB stall status register 2	USBSTLSR2	8	H'FFFE70AA	8
	USB stall status register 3	USBSTLSR3	8	H'FFFE70AB	8
	USB DMA transfer setting register	USBDMAR	8	H'FFFE70B0	8
	USB configuration value register	USBCVR	8	H'FFFE70B4	8
	USB control register	USBCTLR	8	H'FFFE70B8	8
	USB endpoint information register	USBEPIR	8	H'FFFE70C0	8
	USB transceiver test register 0	USBTRNTREG0	8	H'FFFE70D0	8
	USB transceiver test register 1	USBTRNTREG1	8	H'FFFE70D1	8
E-DMAC	E-DMAC mode register	EDMR	32	H'FFFC3000	32
	E-DMAC transmit request register	EDTRR	32	H'FFFC3008	32
	E-DMAC receive request register	EDRRR	32	H'FFFC3010	32
	Transmit descriptor list start address register	TDLAR	32	H'FFFC3018	32
	Receive descriptor list start address register	RDLAR	32	H'FFFC3020	32
	EtherC/E-DMAC status register	EESR	32	H'FFFC3028	32
	EtherC/E-DMAC status interrupt enable register	EESIPR	32	H'FFFC3030	32
	Transmit/receive status copy enable register	TRSCER	32	H'FFFC3038	32
	Receive missed-frame counter register	RMFCR	32	H'FFFC3040	32
	Transmit FIFO threshold register	TFTR	32	H'FFFC3048	32
	FIFO depth register	FDR	32	H'FFFC3050	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
E-DMAC	Receiving method control register	RMCR	32	H'FFFC3058	32
	Transmit FIFO underrun counter register	TFUCR	32	H'FFFC3064	32
	Receive FIFO overflow counter register	RFOCR	32	H'FFFC3068	32
	Receive buffer write address register	RBWAR	32	H'FFFC30C8	32
	Receive descriptor fetch address register	RDFAR	32	H'FFFC30CC	32
	Transmit buffer read address register	TBRAR	32	H'FFFC30D4	32
	Transmit descriptor fetch address register	TDFAR	32	H'FFFC30D8	32
	Flow control start FIFO threshold setting register	FCFTR	32	H'FFFC3070	32
	Transmit interrupt setting register	TRIMD	32	H'FFFC307C	32
	Independent output signal setting register	IOSR	32	H'FFFC306C	32
	E-DMAC operation control register	EDOCR	32	H'FFFC30E4	32
EtherC	EtherC mode register	ECMR	32	H'FFFC3100	32
	EtherC status register	ECSR	32	H'FFFC3110	32
	EtherC interrupt enable register	ECSIPR	32	H'FFFC3118	32
	Receive frame length register	RFLR	32	H'FFFC3108	32
	PHY interface register	PIR	32	H'FFFC3120	32
	MAC address high register	MAHR	32	H'FFFC31C0	32
	MAC address low register	MALR	32	H'FFFC31C8	32
	PHY status register	PSR	32	H'FFFC3128	32
	Transmit retry over counter register	TROCR	32	H'FFFC31D0	32
	Delayed collision detect counter register	CDCR	32	H'FFFC31D4	32
	Lost carrier counter register	LCCR	32	H'FFFC31D8	32
	Carrier not detect counter register	CNDCR	32	H'FFFC31DC	32
	CRC error frame receive counter register	CEFCR	32	H'FFFC31E4	32
	Frame receive error counter register	FRECR	32	H'FFFC31E8	32

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
EtherC	Too-short frame receive counter register	TSFRCR	32	H'FFFC31EC	32
	Too-long frame receive counter register	TLFRCR	32	H'FFFC31F0	32
	Residual-bit frame receive counter register	RFCR	32	H'FFFC31F4	32
	Multicast address frame receive counter register	MAFCR	32	H'FFFC31F8	32
	IPG register	IPGR	32	H'FFFC3150	32
	Automatic PAUSE frame register	APR	32	H'FFFC3154	32
	Manual PAUSE frame register	MPR	32	H'FFFC3158	32
	Automatic PAUSE frame retransmit count register	TPAUSER	32	H'FFFC3164	32
	Random number generation counter upper limit register	RDMLR	32	H'FFFC3140	32
	PAUSE frame receive counter register	RFCF	32	H'FFFC3160	32
	PAUSE frame retransmit counter register	TPAUSECR	32	H'FFFC3168	32
	Broadcast frame receive count register	BCFRR	32	H'FFFC316C	32
ROM/FLD	Flash pin monitor register	FPMON	8	H'FFFFA800	8
	Flash mode register	FMODR	8	H'FFFFA802	8
	Flash access status register	FASTAT	8	H'FFFFA810	8
	Flash access error interrupt enable register	FAEINT	8	H'FFFFA811	8
	ROM MAT select register	ROMMAT	16	H'FFFFA820	8, 16
	FCU RAM enable register	FCURAME	16	H'FFFFA854	8, 16
	Flash status register 0	FSTATR0	8	H'FFFFA900	8, 16
	Flash status register 1	FSTATR1	8	H'FFFFA901	8
	Flash P/E mode entry register	FENTRYR	16	H'FFFFA902	8, 16
	Flash protect register	FPROTR	16	H'FFFFA904	8, 16
	Flash reset register	FRESETR	16	H'FFFFA906	8, 16
	FCU command register	FCMDR	16	H'FFFFA90A	8, 16
	FCU processing switch register	FCPSR	16	H'FFFFA918	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ROM/FLD	FLD blank check control register	EEPBCCNT	16	H'FFFFA91A	8, 16
	Flash P/E status register	FPESTAT	16	H'FFFFA91C	8, 16
	FLD blank check status register	EEPBCSTAT	16	H'FFFFA91E	8, 16
	FLD read enable register 0	EEPRE0	16	H'FFFFA840	8, 16
	FLD program/erase enable register 0	EEPWE0	16	H'FFFFA850	8, 16
	ROM cache control register	RCCR	32	H'FFFC1400	32
	Peripheral clock notification register	PCKAR	16	H'FFFFA938	8, 16
Power-down mode	Standby control register	STBCR	8	H'FFFE0014	8
	Standby control register 2	STBCR2	8	H'FFFE0018	8
	System control register 1	SYSCR1	8	H'FFFE0402	8
	System control register 2	SYSCR2	8	H'FFFE0404	8
	Standby control register 3	STBCR3	8	H'FFFE0408	8
	Standby control register 4	STBCR4	8	H'FFFE040C	8
	Standby control register 5	STBCR5	8	H'FFFE0418	8
	Standby control register 6	STBCR6	8	H'FFFE041C	8
H-UDI	Instruction register	SDIR	16	H'FFFE2000	16

Notes: 1. The access sizes of the WDT registers are different between the read and write to prevent incorrect writing.
2. Use the access size set by the SPLW bit.

32.2 Register Bits

Module Name	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
CPG	FRQCR	—	—	—	—	—	STC[2:0]		
		—	IFC[2:0]			—	PFC[2:0]		
	MCLKCR	—	—	—	—	—	—	MSDIVS[1:0]	
	ACLKCR	—	—	—	—	—	—	ASDIVS[1:0]	
	OSCCR	—	—	—	—	—	OSCSTOP	—	OSCERS
INTC	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE[1:0]		BOVE	—	—	—	—	—
		—	—	—	—	BN[3:0]			
	IPR01	IRQ0				IRQ1			
		IRQ2				IRQ3			
	IPR02	IRQ4				IRQ5			
		IRQ6				IRQ7			
	IPR05	—	—	—	—	—	—	—	—
		ADI0				ADI1			
	IPR06	DMAC0				DMAC1			
		DMAC2				DMAC3			
	IPR07	DMAC4				DMAC5			
		DMAC6				DMAC7			
	IPR08	CMT0				CMT1			
		BSC				WDT			
	IPR09	MTU0				MTU0			
		MTU1				MTU1			
	IPR10	MTU2				MTU2			
		MTU3				MTU3			
	IPR11	MTU4				MTU4			
		MTU5				POE2			

Module Name	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
INTC	IPR12	MTU3S				MTU3S			
		MTU4S				MTU4S			
	IPR13	MTU5S				POE2			
		IIC3				—	—	—	—
	IPR14	—	—	—	—	—	—	—	—
		—	—	—	—	SCIF3			
	IPR15	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	IPR16	SCI0				SCI1			
		SCI2				—	—	—	—
	IPR17	RSPI				SCI4			
		—	—	—	—	—	—	—	—
	IPR18	USB				RCAN-ET			
		USB				USB			
	IPR19	USB				USB			
		EtherC, E-DMAC				—	—	—	—
	USDTENDRR	RXF0	TXF0	RXF1	TXF1	—	—	—	—
		—	—	—	—	—	—	—	—
UBC	BAR_0	BA0_31	BA0_30	BA0_29	BA0_28	BA0_27	BA0_26	BA0_25	BA0_24
		BA0_23	BA0_22	BA0_21	BA0_20	BA0_19	BA0_18	BA0_17	BA0_16
		BA0_15	BA0_14	BA0_13	BA0_12	BA0_11	BA0_10	BA0_9	BA0_8
		BA0_7	BA0_6	BA0_5	BA0_4	BA0_3	BA0_2	BA0_1	BA0_0
	BAMR_0	BAM0_31	BAM0_30	BAM0_29	BAM0_28	BAM0_27	BAM0_26	BAM0_25	BAM0_24
		BAM0_23	BAM0_22	BAM0_21	BAM0_20	BAM0_19	BAM0_18	BAM0_17	BAM0_16
		BAM0_15	BAM0_14	BAM0_13	BAM0_12	BAM0_11	BAM0_10	BAM0_9	BAM0_8
		BAM0_7	BAM0_6	BAM0_5	BAM0_4	BAM0_3	BAM0_2	BAM0_1	BAM0_0
	BBR_0	—	—	UBID0	—	—	CP0[2:0]		
		CD0[1:0]		ID0[1:0]		RW0[1:0]		SZ0[1:0]	
	BAR_1	BA1_31	BA1_30	BA1_29	BA1_28	BA1_27	BA1_26	BA1_25	BA1_24
		BA1_23	BA1_22	BA1_21	BA1_20	BA1_19	BA1_18	BA1_17	BA1_16
		BA1_15	BA1_14	BA1_13	BA1_12	BA1_11	BA1_10	BA1_9	BA1_8
		BA1_7	BA1_6	BA1_5	BA1_4	BA1_3	BA1_2	BA1_1	BA1_0
	BAMR_1	BAM1_31	BAM1_30	BAM1_29	BAM1_28	BAM1_27	BAM1_26	BAM1_25	BAM1_24
		BAM1_23	BAM1_22	BAM1_21	BAM1_20	BAM1_19	BAM1_18	BAM1_17	BAM1_16
		BAM1_15	BAM1_14	BAM1_13	BAM1_12	BAM1_11	BAM1_10	BAM1_9	BAM1_8
		BAM1_7	BAM1_6	BAM1_5	BAM1_4	BAM1_3	BAM1_2	BAM1_1	BAM1_0

Module Name	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
UBC	BBR_1	—	—	UBID1	—	—	CP1[2:0]		
		CD1[1:0]		ID1[1:0]		RW1[1:0]		SZ1[1:0]	
	BAR_2	BA2_31	BA2_30	BA2_29	BA2_28	BA2_27	BA2_26	BA2_25	BA2_24
		BA2_23	BA2_22	BA2_21	BA2_20	BA2_19	BA2_18	BA2_17	BA2_16
		BA2_15	BA2_14	BA2_13	BA2_12	BA2_11	BA2_10	BA2_9	BA2_8
		BA2_7	BA2_6	BA2_5	BA2_4	BA2_3	BA2_2	BA2_1	BA2_0
	BAMR_2	BAM2_31	BAM2_30	BAM2_29	BAM2_28	BAM2_27	BAM2_26	BAM2_25	BAM2_24
		BAM2_23	BAM2_22	BAM2_21	BAM2_20	BAM2_19	BAM2_18	BAM2_17	BAM2_16
		BAM2_15	BAM2_14	BAM2_13	BAM2_12	BAM2_11	BAM2_10	BAM2_9	BAM2_8
		BAM2_7	BAM2_6	BAM2_5	BAM2_4	BAM2_3	BAM2_2	BAM2_1	BAM2_0
	BBR_2	—	—	UBID2	—	—	CP2[2:0]		
		CD2[1:0]		ID2[1:0]		RW2[1:0]		SZ2[1:0]	
	BAR_3	BA3_31	BA3_30	BA3_29	BA3_28	BA3_27	BA3_26	BA3_25	BA3_24
		BA3_23	BA3_22	BA3_21	BA3_20	BA3_19	BA3_18	BA3_17	BA3_16
		BA3_15	BA3_14	BA3_13	BA3_12	BA3_11	BA3_10	BA3_9	BA3_8
		BA3_7	BA3_6	BA3_5	BA3_4	BA3_3	BA3_2	BA3_1	BA3_0
	BAMR_3	BAM3_31	BAM3_30	BAM3_29	BAM3_28	BAM3_27	BAM3_26	BAM3_25	BAM3_24
		BAM3_23	BAM3_22	BAM3_21	BAM3_20	BAM3_19	BAM3_18	BAM3_17	BAM3_16
		BAM3_15	BAM3_14	BAM3_13	BAM3_12	BAM3_11	BAM3_10	BAM3_9	BAM3_8
		BAM3_7	BAM3_6	BAM3_5	BAM3_4	BAM3_3	BAM3_2	BAM3_1	BAM3_0
	BBR_3	—	—	UBID3	—	—	CP3[2:0]		
		CD3[1:0]		ID3[1:0]		RW3[1:0]		SZ3[1:0]	
	BRCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	CKS[1:0]	
		SCMFC0	SCMFC1	SCMFC2	SCMFC3	SCMFD0	SCMFD1	SCMFD2	SCMFD3
		PCB3	PCB2	PCB1	PCB0	—	—	—	—
DTC	DTCERA	DTCERA15	DTCERA14	DTCERA13	DTCERA12	DTCERA11	DTCERA10	DTCERA9	DTCERA8
		DTCERA7	DTCERA6	—	DTCERA4	DTCERA3	DTCERA2	DTCERA1	DTCERA0
	DTCERB	DTCERB15	DTCERB14	DTCERB13	DTCERB12	DTCERB11	DTCERB10	DTCERB9	DTCERB8
		DTCERB7	DTCERB6	DTCERB5	DTCERB4	DTCERB3	DTCERB2	DTCERB1	DTCERB0
	DTCERC	DTCERC15	DTCERC14	DTCERC13	DTCERC12	—	—	—	—
		—	—	—	—	DTCERC3	DTCERC2	DTCERC1	DTCERC0
	DTCERD	DTCERD15	DTCERD14	DTCERD13	DTCERD12	DTCERD11	DTCERD10	DTCERD9	DTCERD8
		DTCERD7	DTCERD6	DTCERD5	DTCERD4	DTCERD3	DTCERD2	—	—

Module Name	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
DTC	DTCERE	DTCERE15	DTCERE14	DTCERE13	DTCERE12	DTCERE11	DTCERE10	DTCERE9	DTCERE8
		DTCERE7	DTCERE6	—	—	—	—	—	—
	DTCCR	—	—	—	RRS	RCHNE	—	—	ERR
	DTCVBR								
						—	—	—	—
		—	—	—	—	—	—	—	—
BSC	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY[1:0]		DMAIW[2]
		DMAIW[1:0]		DMAIWA	—	—	HIZCKIO	HIZMEM	HIZCNT
	CS0BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS1BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS2BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS3BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS4BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		
		—	—	—	—	—	—	—	—
	CS5BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—

Module Name	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	CS6BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS7BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS0WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS0WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1:0]		—	—	BW[1:0]	
		—	—	—	—	—	W[3:1]		
		W[0]	WM	—	—	—	—	—	—
	CS0WCR*6	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	BW[1:0]	
		—	—	—	—	—	W[3:1]		
		W[0]	WM	—	—	—	—	—	—
	CS1WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS2WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3:1]		
		WR[0]	WM	—	—	—	—	—	—

Module Name	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	CS2WCR*3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	A2CL[1]
		A2CL[0]	—	—	—	—	—	—	—
	CS3WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3:1]		
		WR[0]	WM	—	—	—	—	—	—
	CS3WCR*3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	WTRP[1:0]		—	WTRCD[1:0]		—	A3CL[1]
		A3CL[0]	—	—	TRWL[1:0]		—	WTRC[1:0]	
	CS4WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS4WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1:0]		—	—	BW[1:0]	
		—	—	—	SW[1:0]		W[3:1]		
		W[0]	WM	—	—	—	—	HW[1:0]	
	CS5WCR*1	—	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW /BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS6WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS7WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	

Module Name	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	SDCR	—	—	—	—	—	—	—	—
		—	—	—	A2ROW[1:0]		—	A2COL[1:0]	
		—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV
		—	—	—	A3ROW[1:0]		—	A3COL[1:0]	
	RTCSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMF	CMIE	CKS[2:0]			RRC[2:0]		
	RTCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	RTCOR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	BSCEHR	DTLOCK	—	—	—	DTBST	DTSA	—	DTPR
		—	—	—	—	—	—	—	—
DMAC	SAR_0								
	DAR_0								
	DMATCR_0								
	CHCR_0	TC	—	—	RLD	—	—	—	—
		DO	TL	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE

Module Name	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
DMAC	RSAR_0								
	RDAR_0								
	RDMATCR_0								
	SAR_1								
	DAR1								
	DMATCR_1								
	CHCR_1	TC	—	—	RLD	—	—	—	—
		DO	TL	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_1								
	RDAR_1								

Module Name	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
DMAC	RDMATCR_1								
	SAR_2								
	DAR_2								
	DMATCR_2								
	CHCR_2	TC	—	—	RLD	—	—	—	—
		DO	—	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_2								
	RDAR_2								
	RDMATCR_2								
	SAR_3								

Module Name	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
DMAC	DAR_3								
	DMATCR_3								
	CHCR_3	TC	—	—	RLD	—	—	—	—
		DO	—	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_3								
	RDAR_3								
	RDMATCR_3								
	SAR_4								
	DAR_4								
	DMATCR_4								

Module Name	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
DMAC	CHCR_4	TC	—	—	RLD	—	—	—	—
		—	—	—	—	HE	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	TE	DE
	RSAR_4								
	RDAR_4								
	RDMATCR_4								
	SAR_5								
	DAR_5								
	DMATCR_5								
	CHCR_5	TC	—	—	RLD	—	—	—	—
		—	—	—	—	HE	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	TE	DE
	RSAR_5								

Module Name	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
DMAC	RDAR_5								
	RDMATCR_5								
	SAR_6								
	DAR_6								
	DMATCR_6								
	CHCR_6	TC	—	—	RLD	—	—	—	—
		—	—	—	—	HE	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	TE	DE
	RSAR_6								
	RDAR_6								
	RDMATCR_6								

Module Name	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
DMAC	SAR_7								
	DAR_7								
	DMATCR_7								
	CHCR_7	TC	—	—	RLD	—	—	—	—
		—	—	—	—	HE	HIE	—	—
		DM[1:0]		SM[1:0]		RS[3:0]			
		—	—	TB	TS[1:0]		IE	TE	DE
	RSAR_7								
	RDAR_7								
	RDMATCR_7								
	DMAOR	—	—	CMS[1:0]		—	—	PR[1:0]	
		—	—	—	—	—	AE	NMIF	DME
	DMARS0	CH1MID[5:0]						CH1RID[1:0]	
		CH0MID[5:0]						CH0RID[1:0]	
	DMARS1	CH3MID[5:0]						CH3RID[1:0]	
		CH2MID[5:0]						CH2RID[1:0]	
	DMARS2	CH5MID[5:0]						CH5RID[1:0]	
		CH4MID[5:0]						CH4RID[1:0]	

Module Name	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
DMAC	DMARS3	CH7MID[5:0]						CH7RID[1:0]	
		CH6MID[5:0]						CH6RID[1:0]	
MTU2	TCR_0	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_0	—	BFE	BFB	BFA	MD[3:0]			
	TIORH_0	IOB[3:0]				IOA[3:0]			
	TIORL_0	IOD[3:0]				IOC[3:0]			
	TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_0								
	TGRA_0								
	TGRB_0								
	TGRC_0								
	TGRD_0								
	TGRE_0								
	TGRF_0								
	TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE
	TSR2_0	—	—	—	—	—	—	TGFF	TGFE
	TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA
	TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
	TMDR_1	—	—	—	—	MD[3:0]			
	TIOR_1	IOB[3:0]				IOA[3:0]			
	TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
	TCNT_1								
	TGRA_1								

Module Name	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
MTU2	TGRB_1								
	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
	TCR_2	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]		
	TMDR_2	—	—	—	—	MD[3:0]			
	TIOR_2	IOB[3:0]				IOA[3:0]			
	TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
	TCNT_2								
	TGRA_2								
	TGRB_2								
	TCR_3	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_3	—	—	BFB	BFA	MD[3:0]			
	TIORH_3	IOB[3:0]				IOA[3:0]			
	TIORL_3	IOD[3:0]				IOC[3:0]			
	TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3								
	TGRA_3								
	TGRB_3								
	TGRC_3								
	TGRD_3								
	TBTM_3	—	—	—	—	—	—	TTSB	TTSA
	TCR_4	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_4	—	—	BFB	BFA	MD[3:0]			
	TIORH_4	IOB[3:0]				IOA[3:0]			
	TIORL_4	IOD[3:0]				IOC[3:0]			

Module Name	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
MTU2	TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4								
	TGRA_4								
	TGRB_4								
	TGRC_4								
	TGRD_4								
	TBTM_4	—	—	—	—	—	—	TTSB	TTSA
	TADCR	BF[1:0]		—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4								
	TADCORB_4								
	TADCOBRA_4								
	TADCOBRB_4								
	TCRU_5	—	—	—	—	—	—	TPSC[1:0]	
	TCRV_5	—	—	—	—	—	—	TPSC[1:0]	
	TCRW_5	—	—	—	—	—	—	TPSC[1:0]	
	TIORU_5	—	—	—	IOC[4:0]				
	TIORV_5	—	—	—	IOC[4:0]				
	TIORW_5	—	—	—	IOC[4:0]				
	TIER_5	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
	TSR_5	—	—	—	—	—	CMFU5	CMFV5	CMFW5
	TSTR_5	—	—	—	—	—	CSTU5	CSTV5	CSTW5
	TCNTU_5								

Module Name	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
MTU2	TCNTV_5								
	TCNTW_5								
	TGRU_5								
	TGRV_5								
	TGRW_5								
	TCNTCMPCLR	—	—	—	—	—	CMPCLR 5U	CMPCLR 5V	CMPCLR 5W
	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S	SCH4S
	TRWER	—	—	—	—	—	—	—	RWE
	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
	TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCR	—	BDC	N	P	FB	WF	VF	UF
	TCDR								
	TDDR								
	TCNTS								
	TCBR								
	TITCR	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
	TITCNT	—	3ACNT[2:0]			—	4VCNT[2:0]		
	TBTER	—	—	—	—	—	—	BTE[1:0]	
	TDER	—	—	—	—	—	—	—	TDER
	TWCR	CCE	—	—	—	—	—	SCC	WRE
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P

Module Name	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
MTU2S	TCR_3S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_3S	—	—	BFB	BFA	MD[3:0]			
	TIORH_3S	IOB[3:0]				IOA[3:0]			
	TIORL_3S	IOD[3:0]				IOC[3:0]			
	TIER_3S	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3S								
	TGRA_3S								
	TGRB_3S								
	TGRC_3S								
	TGRD_3S								
	TBTM_3S	—	—	—	—	—	—	TTSB	TTSA
	TCR_4S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
	TMDR_4S	—	—	BFB	BFA	MD[3:0]			
	TIORH_4S	IOB[3:0]				IOA[3:0]			
	TIORL_4S	IOD[3:0]				IOC[3:0]			
	TIER_4S	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4S								
	TGRA_4S								
	TGRB_4S								
	TGRC_4S								
	TGRD_4S								
	TBTM_4S	—	—	—	—	—	—	TTSB	TTSA

Module Name	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
MTU2S	TADCRS	BF[1:0]		—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4S								
	TADCORB_4S								
	TADCOBRA_4S								
	TADCOBRB_4S								
	TCRU_5S	—	—	—	—	—	—	TPSC[1:0]	
	TCRV_5S	—	—	—	—	—	—	TPSC[1:0]	
	TCRW_5S	—	—	—	—	—	—	TPSC[1:0]	
	TIORU_5S	—	—	—	IOC[4:0]				
	TIORV_5S	—	—	—	IOC[4:0]				
	TIORW_5S	—	—	—	IOC[4:0]				
	TIER_5S	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
	TSR_5S	—	—	—	—	—	CMFU5	CMFV5	CMFW5
	TSTR_5S	—	—	—	—	—	CSTU5	CSTV5	CSTW5
	TCNTU_5S								
	TCNTV_5S								
	TCNTW_5S								
	TGRU_5S								
	TGRV_5S								
	TGRW_5S								

Module Name	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
MTU2S	TCNT CMPCLRS	—	—	—	—	—	CMPCLR 5U	CMPCLR 5V	CMPCLR 5W
	TSTRS	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYRS	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TRWERS	—	—	—	—	—	—	—	RWE
	TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1S	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
	TOCR2S	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCRS	—	BDC	N	P	FB	WF	VF	UF
	TCDRS								
	TDDRS								
	TCNTSS								
	TCBRS								
	TITCRS	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
	TITCNTS	—	3ACNT[2:0]			—	4VCNT[2:0]		
	TBTERS	—	—	—	—	—	—	BTE[1:0]	
	TDERS	—	—	—	—	—	—	—	TDER
	TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
	TWCRS	CCE	—	—	—	—	—	SCC	WRE
	TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
POE2	ICSR1	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1
		POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[1:0]	
	OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1
		—	—	—	—	—	—	—	—
	ICSR2	—	—	—	POE4F	—	—	—	PIE2
		—	—	—	—	—	—	POE4M[1:0]	

Module Name	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
POE2	OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2
		—	—	—	—	—	—	—	—
	ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3
		—	—	—	—	—	—	POE8M[1:0]	
	SPOER	—	—	—	—	—	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
	POECR1	MTU2 PB4ZE	MTU2 PB3ZE	MTU2 PB2ZE	MTU2 PB1ZE	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
	POECR2	—	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	—	MTU2S SP1CZE	MTU2S SP2CZE	MTU2S SP3CZE
		—	MTU2S SP4CZE	MTU2S SP5CZE	MTU2S SP6CZE	—	MTU2S SP7CZE	MTU2S SP8CZE	MTU2S SP9CZE
CMT	CMSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCSR_0	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1:0]	
	CMCNT_0								
	CMCOR_0								
	CMCSR_1								
		CMF	CMIE	—	—	—	—	CKS[1:0]	
	CMCNT_1								
	CMCOR_1								
WDT	WTCSR	IOVF	WT/$\overline{IT}$	TME	—	—	CKS[2:0]		
	WTCNT								
	WRCSR	WOVF	RSTE	RSTS	—	—	—	—	—
SCI (channel 0)	SCSMR_0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
	SCBRR_0								
	SCSCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	

Module Name	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
SCI (channel 0)	SCTDR_0								
	SCSSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_0								
	SCSDCR_0	—	—	—	—	DIR	—	—	—
	SCSPTR_0	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
SCI (channel 1)	SCSMR_1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
	SCBRR_1								
	SCSCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
	SCTDR_1								
	SCSSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_1								
	SCSDCR_1	—	—	—	—	DIR	—	—	—
	SCSPTR_1	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
SCI (channel 2)	SCSMR_2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
	SCBRR_2								
	SCSCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
	SCTDR_2								
	SCSSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_2								
	SCSDCR_2	—	—	—	—	DIR	—	—	—
	SCSPTR_2	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
SCI (channel 4)	SCSMR_4	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
	SCBRR_4								
	SCSCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
	SCTDR_4								
	SCSSR_4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
	SCRDR_4								
	SCSDCR_4	—	—	—	—	DIR	—	—	—
	SCSPTR_4	EIO	—	—	—	SPB1IO	SPB1DT	—	SPB0DT
SCIF	SCSMR_3	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR_3								
	SCSCR_3	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	

Module Name	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
SCIF	SCFTDR_3								
	SCFSR_3	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_3								
	SCFCR_3	—	—	—	—	—	—	—	—
		RTRG[1:0]		TTRG[1:0]		—	TFRST	RFRST	LOOP
	SCFDR_3	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_3	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSEMR_3	ABCS	—	—	—	—	—	—	—
RSPI	SPCR	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	—	SPMS
	SSLP	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
	SPPCR	—	—	MOIFE	MOIFV	—	SPOM	—	SPLP
	SPSR	SPRF	—	SPTEF	—	—	MODF	MIDLE	OVRF
	SPDR	SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24
		SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
		SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8
		SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
	SPSCR	—	—	—	—	—	SPSLN2	SPSLN1	SPSLN0
	SPSSR	—	—	SPECM1	SPECM0	—	—	SPCP1	SPCP0
	SPBR	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
	SPDCR	—	—	SPLW	SPRDTD	—	—	SPFC1	SPFC0
	SPCKD	—	—	—	—	—	SCKDL2	SCKDL1	SCKDL0
	SSLND	—	—	—	—	—	SLNDL2	SLNDL1	SLNDL0
	SPND	—	—	—	—	—	SPNDL2	SPNDL1	SPNDL0
	SPCMD0	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
	SPCMD1	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
	SPCMD2	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA
	SPCMD3	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB3	SPB2	SPB1	SPB0
		SSLKP	SSLA2	SSLA1	SSLA0	BRDV1	BRDV0	CPOL	CPHA

Module Name	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
IIC3	ICCR1	ICE	RCVD	MST	TRS	CKS[3:0]			
	ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
	ICMR	MLS	—	—	—	BCWP	BC[2:0]		
	ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR	SVA[6:0]							FS
	ICDRT								
	ICDRR								
	NF2CYC	—	—	—	—	—	—	—	NF2CYC
ADC	ADCR_0	ADST	ADCS	ACE	ADIE	—	—	TRGE	EXTRG
	ADSR_0	—	—	—	—	—	—	—	ADF
	ADSTRGR_0	—	STR6	STR5	STR4	STR3	STR2	STR1	STR0
	ADANSR_0	—	—	—	—	ANS3	ANS2	ANS1	ANS0
	ADBYPSCR_0	—	—	—	—	—	—	—	SH
	ADDR0	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR1	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR2	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR3	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADCR_1	ADST	ADCS	ACE	ADIE	—	—	TRGE	EXTRG
	ADSR_1	—	—	—	—	—	—	—	ADF
	ADSTRGR_1	—	STR6	STR5	STR4	STR3	STR2	STR1	STR0
	ADANSR_1	—	—	—	—	ANS3	ANS2	ANS1	ANS0
	ADBYPSCR_1	—	—	—	—	—	—	—	—
	ADDR4	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR5	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR6	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR7	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

Module Name	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
RCAN-ET	MCR	MCR15	MCR14	—	—	—	TST[2:0]		
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1	TSG1[3:0]				—	TSG2[2:0]		
		—	—	SJW[1:0]		—	—	—	BSP
	BCR0	—	—	—	—	—	—	—	—
		BRP[7:0]							
	IRR	—	—	IRR13	IRR12	—	—	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
	TXPR1, 0	TXPR1[15:8]							
		TXPR1[7:0]							
		TXPR0[15:8]							
		TXPR0[7:1]							—
	TXCR0	TXCR0[15:8]							
		TXCR0[7:1]							—
	TXACK0	TXACK0[15:8]							
		TXACK0[7:1]							—
	ABACK0	ABACK0[15:8]							
		ABACK0[7:1]							—
	RXPR0	RXPR0[15:8]							
		RXPR0[7:0]							
	RFPR0	RFPR0[15:8]							
		RFPR0[7:0]							
	MBIMR0	MBIMR0[15:8]							
		MBIMR0[7:0]							
	UMSR0	UMSR0[15:8]							
		UMSR0[7:0]							

Module Name	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
RCAN-ET (MCR15 = 1)	MB[0]. CONTROL0H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	
RCAN-ET (MCR15 = 0)	MB[0]. CONTROL0H	—	STDID[10:4]						
		STDID[3:0]				RTR	IDE	EXTID[17:16]	
RCAN-ET	MB[0]. CONTROL0L	EXTID[15:8]							
		EXTID[7:0]							
RCAN-ET (MCR15 = 1)	MB[0]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
RCAN-ET (MCR15 = 0)	MB[0]. LAFMH	—	STDID_LAFM[10:4]						
		STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]	
RCAN-ET	MB[0]. LAFML	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[0]. MSG_DATA [0]	MSG_DATA_0							
	MB[0]. MSG_DATA [1]	MSG_DATA_1							
	MB[0]. MSG_DATA [2]	MSG_DATA_2							
	MB[0]. MSG_DATA [3]	MSG_DATA_3							
	MB[0]. MSG_DATA [4]	MSG_DATA_4							
	MB[0]. MSG_DATA[5]	MSG_DATA_5							
	MB[0]. MSG_DATA [6]	MSG_DATA_6							
	MB[0]. MSG_DATA [7]	MSG_DATA_7							
	MB[0]. CONTROL1H	—	—	NMC	—	—	MBC[2:0]		
	MB[0]. CONTROL1L	—	—	—	—	DLC[3:0]			
RCAN-ET (MCR15 = 1)	MB[1]. CONTROL0H	IDE	RTR	—	STDID[10:6]				
		STDID[5:0]						EXTID[17:16]	

Module Name	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
RCAN-ET (MCR15 = 0)	MB[1]. CONTROL0H	—	STDID[10:4]						
		STDID[3:0]				RTR	IDE	EXTID[17:16]	
RCAN-ET	MB[1]. CONTROL0L	EXTID[15:8]							
		EXTID[7:0]							
RCAN-ET (MCR15 = 1)	MB[1]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]				
		STDID_LAFM[5:0]						EXTID_LAFM[17:16]	
RCAN-ET (MCR15 = 0)	MB[1]. LAFMH	—	STDID_LAFM[10:4]						
		STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]	
RCAN-ET	MB[1]. LAFML	EXTID_LAFM[15:8]							
		EXTID_LAFM[7:0]							
	MB[1]. MSG_DATA[0]	MSG_DATA0							
	MB[1]. MSG_DATA[1]	MSG_DATA1							
	MB[1]. MSG_DATA[2]	MSG_DATA2							
	MB[1]. MSG_DATA[3]	MSG_DATA3							
	MB[1]. MSG_DATA[4]	MSG_DATA4							
	MB[1]. MSG_DATA[5]	MSG_DATA5							
	MB[1]. MSG_DATA[6]	MSG_DATA6							
	MB[1]. MSG_DATA[7]	MSG_DATA7							

Module Name	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
RCAN-ET	MB[1]. CONTROL1H	—	—	NMC	ATX	DART	MBC[2:0]		
	MB[1]. CONTROL1L	—	—	—	—	DLC[3:0]			
	MB[2].	Same bit configuration as MB[1]							
	MB[3].	Same bit configuration as MB[1]							
	↓	(Ditto)							
	MB[13].	Same bit configuration as MB[1]							
	MB[14].	Same bit configuration as MB[1]							
	MB[15].	Same bit configuration as MB[1]							
PFC	PAIORH	—	—	—	—	—	—	—	—
		—	—	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR
	PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR
		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR
	PACRH2	—	—	—	—	—	—	—	—
		—	PA21MD[2:0]			—	PA20MD[2:0]		
	PACRH1	—	PA19MD[2:0]			—	PA18MD[2:0]		
		—	PA17MD[2:0]			—	PA16MD[2:0]		
	PACRL4	—	PA15MD[2:0]			—	PA14MD[2:0]		
		—	PA13MD[2:0]			—	PA12MD[2:0]		
	PACRL3	—	PA11MD[2:0]			—	PA10MD[2:0]		
		—	PA9MD[2:0]			—	PA8MD[2:0]		
	PACRL2	—	PA7MD[2:0]			—	PA6MD[2:0]		
		—	PA5MD[2:0]			—	PA4MD[2:0]		
	PACRL1	—	PA3MD[2:0]			—	PA2MD[2:0]		
		—	PA1MD[2:0]			—	PA0MD[2:0]		
	PAPCRH	—	—	—	—	—	—	—	—
		—	—	PA21PCR	PA20PCR	PA19PCR	PA18PCR	PA17PCR	PA16PCR
	PAPCRL	PA15PCR	PA14PCR	PA13PCR	PA12PCR	PA11PCR	PA10PCR	PA9PCR	PA8PCR
		PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR

Module Name	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
PFC	PBIORL	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
	PBCRL4	—	PB15MD[2:0]			—	PB14MD[2:0]		
		—	PB13MD[2:0]			—	PB12MD[2:0]		
	PBCRL3	—	PB11MD[2:0]			—	PB10MD[2:0]		
		—	PB9MD[2:0]			—	PB8MD[2:0]		
	PBCRL2	—	PB7MD[2:0]			—	PB6MD[2:0]		
		—	PB5MD[2:0]			—	PB4MD[2:0]		
	PBCRL1	—	PB3MD[2:0]			—	PB2MD[2:0]		
		—	PB1MD[2:0]			—	PB0MD[2:0]		
	PBPCRL	PB15PCR	PB14PCR	PB13PCR	PB12PCR	PB11PCR	PB10PCR	PB9PCR	PB8PCR
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
	PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCCRL4	—	PC15MD[2:0]			—	PC14MD[2:0]		
		—	PC13MD[2:0]			—	PC12MD[2:0]		
	PCCRL3	—	PC11MD[2:0]			—	PC10MD[2:0]		
		—	PC9MD[2:0]			—	PC8MD[2:0]		
	PCCRL2	—	PC7MD[2:0]			—	PC6MD[2:0]		
		—	PC5MD[2:0]			—	PC4MD[2:0]		
	PCCRL1	—	PC3MD[2:0]			—	PC2MD[2:0]		
		—	PC1MD[2:0]			—	PC0MD[2:0]		
	PCPCRL	PC15PCR	PC14PCR	PC13PCR	PC12PCR	PC11PCR	PC10PCR	PC19PCR	PC8PCR
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
	PDIORH	PD31IOR	PD30IOR	PD29IOR	PD28IOR	PD27IOR	PD26IOR	PD25IOR	PD24IOR
		PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IOR
	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	PDCRH4	—	PD31MD[2:0]			—	PD30MD[2:0]		
		—	PD29MD[2:0]			—	PD28MD[2:0]		
	PDCRH3	—	PD27MD[2:0]			—	PD26MD[2:0]		
		—	PD25MD[2:0]			—	PD24MD[2:0]		

Module Name	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
PFC	PDCRH2	—	PD23MD[2:0]			—	PD22MD[2:0]		
		—	PD21MD[2:0]			—	PD20MD[2:0]		
	PDCRH1	—	PD19MD[2:0]			—	PD18MD[2:0]		
		—	PD17MD[2:0]			—	PD16MD[2:0]		
	PDCRL4	—	PD15MD[2:0]			—	PD14MD[2:0]		
		—	PD13MD[2:0]			—	PD12MD[2:0]		
	PDCRL3	—	PD11MD[2:0]			—	PD10MD[2:0]		
		—	PD9MD[2:0]			—	PD8MD[2:0]		
	PDCRL2	—	PD7MD[2:0]			—	PD6MD[2:0]		
		—	PD5MD[2:0]			—	PD4MD[2:0]		
	PDCRL1	—	PD3MD[2:0]			—	PD2MD[2:0]		
		—	PD1MD[2:0]			—	PD0MD[2:0]		
	PDPCRH	PD31PCR	PD30PCR	PD29PCR	PD28PCR	PD27PCR	PD26PCR	PD25PCR	PD24PCR
		PD23PCR	PD22PCR	PD21PCR	PD20PCR	PD19PCR	PD18PCR	PD17PCR	PD16PCR
	PDPCRL	PD15PCR	PD14PCR	PD13PCR	PD12PCR	PD11PCR	PD10PCR	PD9PCR	PD8PCR
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
	PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR
		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR
	PECRL4	—	PE15MD[2:0]			—	PE14MD[2:0]		
		—	PE13MD[2:0]			—	PE12MD[2:0]		
	PECRL3	—	PE11MD[2:0]			—	PE10MD[2:0]		
		—	PE9MD[2:0]			—	PE8MD[2:0]		
	PECRL2	—	PE7MD[2:0]			—	PE6MD[2:0]		
		—	PE5MD[2:0]			—	PE4MD[2:0]		
	PECRL1	—	PE3MD[2:0]			—	PE2MD[2:0]		
		—	PE1MD[2:0]			—	PE0MD[2:0]		
	HCPCR	—	—	—	—	—	—	—	—
		—	—	—	—	MZIZDH	MZIZDL	MZIZEH	MZIZEL
	IFCR	—	—	—	—	—	—	—	—
		—	—	—	—	IRQMD3	IRQMD2	IRQMD1	IRQMD0
	PEPCRL	PE15PCR	PE14PCR	PE13PCR	PE12PCR	PE11PCR	PE10PCR	PE9PCR	PE8PCR
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR

Module Name	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
PFC	PDACKCR	—	—	—	—	—	—	—	—
		—	—	—	—	DACK3TMG	DACK2TMG	DACK1TMG	DACK0TMG
I/O port	PADRH	—	—	—	—	—	—	—	—
		—	—	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PAPRH	—	—	—	—	—	—	—	—
		—	—	PA21PR	PA20PR	PA19PR	PA18PR	PA17PR	PA16PR
	PAPRL	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR
		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
	PBDRL	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPRL	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPRL	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PDDRH	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR
		PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR
	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
	PDPRH	PD31PR	PD30PR	PD29PR	PD28PR	PD27PR	PD26PR	PD25PR	PD24PR
		PD23PR	PD22PR	PD21PR	PD20PR	PD19PR	PD18PR	PD17PR	PD16PR
	PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
	PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPRL	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR
		PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PFDRL	—	—	—	—	—	—	—	—
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR

Module Name	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
USB	USBIFR0	BRST	CFDN	—	—	SETC	SETI	VBUSMN	VBUSF
	USBIFR1	—	—	—	SOF	SETUPTS	EP0oTS	EP0iTR	EP0iTS
	USBIFR2	—	—	EP3TR	EP3TS	EP2TR	EP2 EMPTY	EP2 ALLEMP	EP1FULL
	USBIFR3	—	—	EP6TR	EP6TS	EP5TR	EP5 EMPTY	EP5 ALLEMP	EP4FULL
	USBIFR4	—	—	EP9TR	EP9TS	EP8TR	EP8 EMPTY	—	EP7FULL
	USBIER0	BRSTE	CFDFN	—	—	SETCE	SETIE	—	VBUSFE
	USBIER1	—	—	—	SOFE	SETUPTSE	EP0oTSE	EP0iTRE	EP0iTSE
	USBIER2	—	—	EP3TRE	EP3TSE	EP2TRE	EP2 EMPTYE	EP2 ALLEMPE	EP1FULLE
	USBIER3	—	—	EP6TRE	EP6TSE	EP5TRE	EP5 EMPTYE	EP5 ALLEMPE	EP4FULLE
	USBIER4	—	—	EP9TRE	EP9TSE	EP8TRE	EP8 EMPTYE	—	EP7FULLE
	USBISR0	BRSTS	CFDNS	—	—	SETCS	SETIS	—	VBUSFS
	USBISR1	—	—	—	SOFS	SETUPTSS	EP0oTSS	EP0iTRS	EP0iTSS
	USBISR2	—	—	EP3TRS	EP3TSS	EP2TRS	EP2 EMPTYS	EP2 ALLEMPS	EP1FULLS
	USBISR3	—	—	EP6TRS	EP6TSS	EP5TRS	EP5 EMPTYS	EP5 ALLEMPS	EP4FULLS
	USBISR4	—	—	EP9TRS	EP9TSS	EP8TRS	EP8 EMPTYS	—	EP7FULLS
	USBEPDR0i	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR0o	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR0s	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR1	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR2	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR3	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR4	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR5	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR6	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR7	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR8	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPDR9	D7	D6	D5	D4	D3	D2	D1	D0
	USBEPSZ0o	—	—	—	D4	D3	D2	D1	D0
	USBEPSZ1	—	D6	D5	D4	D3	D2	D1	D0
	USBEPSZ4	—	D6	D5	D4	D3	D2	D1	D0

Module Name	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
USB	USBEPSZ7	—	D6	D5	D4	D3	D2	D1	D0
	USBDASTS0	—	—	—	—	—	—	—	EP0iDE
	USBDASTS1	—	—	—	—	—	EP3DE	EP2DE	—
	USBDASTS2	—	—	—	—	—	EP6DE	EP5DE	—
	USBDASTS3	—	—	—	—	—	EP9DE	EP8DE	—
	USBTRG0	—	—	—	—	—	EP0sRDFN	EP0oRDFN	EP0iPKTE
	USBTRG1	—	—	—	—	—	EP3PKTE	EP2PKTE	EP1RDFN
	USBTRG2	—	—	—	—	—	EP6PKTE	EP5PKTE	EP4RDFN
	USBTRG3	—	—	—	—	—	EP9PKTE	EP8PKTE	EP7RDFN
	USBFCLR0	—	—	—	—	—	—	EP0oCLR	EP0iCLR
	USBFCLR1	—	—	—	—	—	EP3CLR	EP2CLR	EP1CLR
	USBFCLR2	—	—	—	—	—	EP6CLR	EP5CLR	EP4CLR
	USBFCLR3	—	—	—	—	—	EP9CLR	EP8CLR	EP7CLR
	USBEPSTL0	—	—	—	EP0STLC	—	—	—	EP0STLS
	USBEPSTL1	—	EP3STLC	EP2STLC	EP1STLC	—	EP3STLS	EP2STLS	EP1STLS
	USBEPSTL2	—	EP6STLC	EP5STLC	EP4STLC	—	EP6STLS	EP5STLS	EP4STLS
	USBEPSTL3	—	EP9STLC	EP8STLC	EP7STLC	—	EP9STLS	EP8STLS	EP7STLS
	USBSTLSR1	—	EP3ASCE	EP2ASCE	EP1ASCE	—	EP3STLST	EP2STLST	EP1STLST
	USBSTLSR2	—	EP6ASCE	EP5ASCE	EP4ASCE	—	EP6STLST	EP5STLST	EP4STLST
	USBSTLSR3	—	EP9ASCE	EP8ASCE	EP7ASCE	—	EP9STLST	EP8STLST	EP7STLST
	USBDMAR	—	—	—	EP5DMAE	EP4DMAE	—	EP2DMAE	EP1DMAE
	USBCVR	CNFV1	CNFV0	INTV1	INTV0	—	ALTV2	ALTV1	ALTV0
	USBCTLR	—	—	—	—	—	—	EP0ASCE	PRTRST
	USBEPIR	D7	D6	D5	D4	D3	D2	D1	D0
	USBTRNTREG0	PTSTE	—	—	—	SUSPEND	txenl	txse0	txdata
	USBTRNTREG1	—	—	—	—	—	xver_data	dpls	dmns
E-DMAC	EDMR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	DE	DL1	DL0	—	—	—	SWR
	EDTRR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	TR

Module Name	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
E-DMAC	EDRRR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	RR
	TDLAR	TDLA31	TDLA30	TDLA29	TDLA28	TDLA27	TDLA26	TDLA25	TDLA24
		TDLA23	TDLA22	TDLA21	TDLA20	TDLA19	TDLA18	TDLA17	TDLA16
		TDLA15	TDLA14	TDLA13	TDLA12	TDLA11	TDLA10	TDLA9	TDLA8
		TDLA7	TDLA6	TDLA5	TDLA4	TDLA3	TDLA2	TDLA1	TDLA0
	RDLAR	RDLA31	RDLA30	RDLA29	RDLA28	RDLA27	RDLA26	RDLA25	RDLA24
		RDLA23	RDLA22	RDLA21	RDLA20	RDLA19	RDLA18	RDLA17	RDLA16
		RDLA15	RDLA14	RDLA13	RDLA12	RDLA11	RDLA10	RDLA9	RDLA8
		RDLA7	RDLA6	RDLA5	RDLA4	RDLA3	RDLA2	RDLA1	RDLA0
	EESR	—	TWB	—	—	—	TABT	RABT	RFCOF
		ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
		—	—	—	—	CND	DLC	CD	TRO
		RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
	EESIPR	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP
		ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
		—	—	—	—	CNDIP	DLCIP	CDIP	TROIP
		RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
	TRSCER	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CNDCE	DLCCE	CDCE	TROCE
		RMAFCE	—	—	RRFCE	RTLFCE	RTSFCE	PRECE	CERFCE
	RMFCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		MFC15	MFC14	MFC13	MFC12	MFC11	MFC10	MFC9	MFC8
		MFC7	MFC6	MFC5	MFC4	MFC3	MFC2	MFC1	MFC0
	TFTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	TFT10	TFT9	TFT8
		TFT7	TFT6	TFT5	TFT4	TFT3	TFT2	TFT1	TFT0
	FDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	TFD4	TFD3	TFD2	TFD1	TFD0
		—	—	—	RFD4	RFD3	RFD2	RFD1	RFD0

Module Name	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
E-DMAC	RMCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	RNC	RNR
	TFUCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		UNDER15	UNDER14	UNDER13	UNDER12	UNDER11	UNDER10	UNDER9	UNDER8
		UNDER7	UNDER6	UNDER5	UNDER4	UNDER3	UNDER2	UNDER1	UNDER0
	RFOCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		OVER15	OVER14	OVER13	OVER12	OVER11	OVER10	OVER9	OVER8
		OVER7	OVER6	OVER5	OVER4	OVER3	OVER2	OVER1	OVER0
	RBWAR	RBWA31	RBWA30	RBWA29	RBWA28	RBWA27	RBWA26	RBWA25	RBWA24
		RBWA23	RBWA22	RBWA21	RBWA20	RBWA19	RBWA18	RBWA17	RBWA16
		RBWA15	RBWA14	RBWA13	RBWA12	RBWA11	RBWA10	RBWA9	RBWA8
		RBWA7	RBWA6	RBWA5	RBWA4	RBWA3	RBWA2	RBWA1	RBWA0
	RDFAR	RDFA31	RDFA30	RDFA29	RDFA28	RDFA27	RDFA26	RDFA25	RDFA24
		RDFA23	RDFA22	RDFA21	RDFA20	RDFA19	RDFA18	RDFA17	RDFA16
		RDFA15	RDFA14	RDFA13	RDFA12	RDFA11	RDFA10	RDFA9	RDFA8
		RDFA7	RDFA6	RDFA5	RDFA4	RDFA3	RDFA2	RDFA1	RDFA0
	TBRAR	TBRA31	TBRA30	TBRA29	TBRA28	TBRA27	TBRA26	TBRA25	TBRA24
		TBRA23	TBRA22	TBRA21	TBRA20	TBRA19	TBRA18	TBRA17	TBRA16
		TBRA15	TBRA14	TBRA13	TBRA12	TBRA11	TBRA10	TBRA9	TBRA8
		TBRA7	TBRA6	TBRA5	TBRA4	TBRA3	TBRA2	TBRA1	TBRA0
	TDFAR	TDFA31	TDFA30	TDFA29	TDFA28	TDFA27	TDFA26	TDFA25	TDFA24
		TDFA23	TDFA22	TDFA21	TDFA20	TDFA19	TDFA18	TDFA17	TDFA16
		TDFA15	TDFA14	TDFA13	TDFA12	TDFA11	TDFA10	TDFA9	TDFA8
		TDFA7	TDFA6	TDFA5	TDFA4	TDFA3	TDFA2	TDFA1	TDFA0
	FCFTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	RFFO2	RFFO1	RFFO0
		—	—	—	—	—	—	—	—
		—	—	—	—	—	RFDO2	RFDO1	RFDO0
	TRIMD	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	TIM	—	—	—	TIS

Module Name	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
E-DMAC	IOSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ELB
	EDOCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	FEC	AEC	EDH	NMIE
EtherC	ECMR	—	—	—	—	—	—	—	—
		—	—	—	TPC	ZPF	PFR	RXF	TXF
		—	—	—	PRCEF	—	—	MPDE	—
		—	PE	TE	—	ILB	—	DM	PRM
	ECSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
	ECSIPR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
	RFLR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	RFL11	RFL10	RFL9	RFL8
		RFL7	RFL6	RFL5	RFL4	RFL3	RFL2	RFL1	RFL0
	PIR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	MDI	MDO	MMD	MDC
	MAHR	MA47	MA46	MA45	MA44	MA43	MA42	MA41	MA40
		MA39	MA38	MA37	MA36	MA35	MA34	MA33	MA32
		MA31	MA30	MA29	MA28	MA27	MA26	MA25	MA24
		MA23	MA22	MA21	MA20	MA19	MA18	MA17	MA16
	MALR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8
		MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0

Module Name	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	PSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LMON
	TROCR	TROC31	TROC30	TROC29	TROC28	TROC27	TROC26	TROC25	TROC24
		TROC23	TROC22	TROC21	TROC20	TROC19	TROC18	TROC17	TROC16
		TROC15	TROC14	TROC13	TROC12	TROC11	TROC10	TROC9	TROC8
		TROC7	TROC6	TROC5	TROC4	TROC3	TROC2	TROC1	TROC0
	CDCR	COSDC31	COSDC30	COSDC29	COSDC28	COSDC27	COSDC26	COSDC25	COSDC24
		COSDC23	COSDC22	COSDC21	COSDC20	COSDC19	COSDC18	COSDC17	COSDC16
		COSDC15	COSDC14	COSDC13	COSDC12	COSDC11	COSDC10	COSDC9	COSDC8
		COSDC7	COSDC6	COSDC5	COSDC4	COSDC3	COSDC2	COSDC1	COSDC0
	LCCR	LCC31	LCC30	LCC29	LCC28	LCC27	LCC26	LCC25	LCC24
		LCC23	LCC22	LCC21	LCC20	LCC19	LCC18	LCC17	LCC16
		LCC15	LCC14	LCC13	LCC12	LCC11	LCC10	LCC9	LCC8
		LCC7	LCC6	LCC5	LCC4	LCC3	LCC2	LCC1	LCC0
	CNDCR	CNDC31	CNDC30	CNDC29	CNDC28	CNDC27	CNDC26	CNDC25	CNDC24
		CNDC23	CNDC22	CNDC21	CNDC20	CNDC19	CNDC18	CNDC17	CNDC16
		CNDC15	CNDC14	CNDC13	CNDC12	CNDC11	CNDC10	CNDC9	CNDC8
		CNDC7	CNDC6	CNDC5	CNDC4	CNDC3	CNDC2	CNDC1	CNDC0
	CEFCR	CEFC31	CEFC30	CEFC29	CEFC28	CEFC27	CEFC26	CEFC25	CEFC24
		CEFC23	CEFC22	CEFC21	CEFC20	CEFC19	CEFC18	CEFC17	CEFC16
		CEFC15	CEFC14	CEFC13	CEFC12	CEFC11	CEFC10	CEFC9	CEFC8
		CEFC7	CEFC6	CEFC5	CEFC4	CEFC3	CEFC2	CEFC1	CEFC0
	FRECR	FREC31	FREC30	FREC29	FREC28	FREC27	FREC26	FREC25	FREC24
		FREC23	FREC22	FREC21	FREC20	FREC19	FREC18	FREC17	FREC16
		FREC15	FREC14	FREC13	FREC12	FREC11	FREC10	FREC9	FREC8
		FREC7	FREC6	FREC5	FREC4	FREC3	FREC2	FREC1	FREC0
	TSFRCR	TSFC31	TSFC30	TSFC29	TSFC28	TSFC27	TSFC26	TSFC25	TSFC24
		TSFC23	TSFC22	TSFC21	TSFC20	TSFC19	TSFC18	TSFC17	TSFC16
		TSFC15	TSFC14	TSFC13	TSFC12	TSFC11	TSFC10	TSFC9	TSFC8
		TSFC7	TSFC6	TSFC5	TSFC4	TSFC3	TSFC2	TSFC1	TSFC0
	TLFRCR	TLFC31	TLFC30	TLFC29	TLFC28	TLFC27	TLFC26	TLFC25	TLFC24
		TLFC23	TLFC22	TLFC21	TLFC20	TLFC19	TLFC18	TLFC17	TLFC16
		TLFC15	TLFC14	TLFC13	TLFC12	TLFC11	TLFC10	TLFC9	TLFC8
		TLFC7	TLFC6	TLFC5	TLFC4	TLFC3	TLFC2	TLFC1	TLFC0

Module Name	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	RFC31	RFC30	RFC29	RFC28	RFC27	RFC26	RFC25	RFC24
		RFC23	RFC22	RFC21	RFC20	RFC19	RFC18	RFC17	RFC16
		RFC15	RFC14	RFC13	RFC12	RFC11	RFC10	RFC9	RFC8
		RFC7	RFC6	RFC5	RFC4	RFC3	RFC2	RFC1	RFC0
	MAFCR	MAFC31	MAFC30	MAFC29	MAFC28	MAFC27	MAFC26	MAFC25	MAFC24
		MAFC23	MAFC22	MAFC21	MAFC20	MAFC19	MAFC18	MAFC17	MAFC16
		MAFC15	MAFC14	MAFC13	MAFC12	MAFC11	MAFC10	MAFC9	MAFC8
		MAFC7	MAFC6	MAFC5	MAFC4	MAFC3	MAFC2	MAFC1	MAFC0
	IPGR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	IPG4	IPG3	IPG2	IPG1	IPG0
	APR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		AP15	AP14	AP13	AP12	AP11	AP10	AP9	AP8
		AP7	AP6	AP5	AP4	AP3	AP2	AP1	AP0
	MPR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		MP15	MP14	MP13	MP12	MP11	MP10	MP9	MP8
		MP7	MP6	MP5	MP4	MP3	MP2	MP1	MP0
	TPAUSER	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TPAUSE15	TPAUSE14	TPAUSE13	TPAUSE12	TPAUSE11	TPAUSE10	TPAUSE9	TPAUSE8
		TPAUSE7	TPAUSE6	TPAUSE5	TPAUSE4	TPAUSE3	TPAUSE2	TPAUSE1	TPAUSE0
	RDMLR	—	—	—	—	—	—	—	—
		—	—	—	—	RMD19	RMD18	RMD17	RMD16
		RMD15	RMD14	RMD13	RMD12	RMD11	RMD10	RMD9	RMD8
		RMD7	RMD6	RMD5	RMD4	RMD3	RMD2	RMD1	RMD0
	RFCF	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		RPAUSE7	RPAUSE6	RPAUSE5	RPAUSE4	RPAUSE3	RPAUSE2	RPAUSE1	RPAUSE0
	TPAUSECR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		TXP7	TXP6	TXP5	TXP4	TXP3	TXP2	TXP1	TXP0

Module Name	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	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		BCF15	BCF14	BCF13	BCF12	BCF11	BCF10	BCF9	BCF8
		BCF7	BCF6	BCF5	BCF4	BCF3	BCF2	BCF1	BCF0
ROM/FLD	FPMON	FWE	—	—	—	—	—	—	—
	FMODR	—	—	—	FRDMD	—	—	—	—
	FASTAT	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
	FAEINT	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEE
	ROMMAT	KEY							
		—	—	—	—	—	—	—	ROMSEL
	FCURAME	KEY							
		—	—	—	—	—	—	—	FCRME
	FSTATR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
	FSTATR1	FCUERR	—	—	FLOCKST	—	—	FRDTCT	FRCRCT
	FENTRYR	FKEY							
		FENTRYD	—	—	—	—	—	—	FENTRY0
	FPROTR	FPKEY							
		—	—	—	—	—	—	—	FPROTCN
	FRESETR	FPKEY							
		—	—	—	—	—	—	—	FRESET
	FCMDR	CMDR							
		PCMDR							
	FCPSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ESUSPMD
	EEPBCCNT	—	—	—	BCADR				
		BCADR					—	—	BCSIZE
	FPESTAT	—	—	—	—	—	—	—	—
		PEERRST							
	EEPBCSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BCST
	EEPRE0	KEY							
		—	—	—	—	DBRE03	DBRE02	DBRE01	DBRE00
	EEPWE0	KEY							
		—	—	—	—	DBWE03	DBWE02	DBWE01	DBWE00

Module Name	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
ROM/FLD	RCCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	RCF	—	—	—
	PCKAR	—	—	—	—	—	—	—	—
		PCKA							
Power-down mode	STBCR	STBY	—	—	—	—	—	—	—
	STBCR2	MSTP10	MSTP9	MSTP8	—	—	—	MSTP4	—
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	STBCR3	HIZ	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	—	MSTP30
	STBCR4	—	—	—	MSTP44	—	MSTP42	—	MSTP40
	STBCR5	MSTP57	MSTP56	MSTP55	—	MSTP53	MSTP52	—	MSTP50
	STBCR6	USBSEL	MSTP66	USBCLK	MSTP64	—	—	—	—
H-UDI	SDIR	TI[3:0]				—	—	—	—
		—	—	—	—	—	—	—	—

Notes: 1. When normal space, SRAM with byte selection, or MPX-I/O is the memory type
2. When burst ROM (clocked asynchronous) is the memory type
3. When SDRAM is the memory type
4. When burst ROM (clocked synchronous) is the memory type

32.3 Register States in Each Operating Mode

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
CPG	FRQCR	Initialized*[1]	Retained	Retained	—	Retained
	MCLKCR	Initialized	Retained	Retained	—	Retained
	ACLKCR	Initialized	Retained	Retained	—	Retained
	OSCCR	Initialized	Retained	Retained	—	Retained
INTC	ICR0	Initialized	Retained	Retained	—	Retained
	ICR1	Initialized	Retained	Retained	—	Retained
	IRQRR	Initialized	Retained	Retained	—	Retained
	IBCR	Initialized	Retained	Retained	—	Retained
	IBNR	Initialized	Retained*[2]	Retained	—	Retained
	IPR01	Initialized	Retained	Retained	—	Retained
	IPR02	Initialized	Retained	Retained	—	Retained
	IPR05	Initialized	Retained	Retained	—	Retained
	IPR06	Initialized	Retained	Retained	—	Retained
	IPR07	Initialized	Retained	Retained	—	Retained
	IPR08	Initialized	Retained	Retained	—	Retained
	IPR09	Initialized	Retained	Retained	—	Retained
	IPR10	Initialized	Retained	Retained	—	Retained
	IPR11	Initialized	Retained	Retained	—	Retained
	IPR12	Initialized	Retained	Retained	—	Retained
	IPR13	Initialized	Retained	Retained	—	Retained
	IPR14	Initialized	Retained	Retained	—	Retained
	IPR15	Initialized	Retained	Retained	—	Retained
	IPR16	Initialized	Retained	Retained	—	Retained
	IPR17	Initialized	Retained	Retained	—	Retained
	IPR18	Initialized	Retained	Retained	—	Retained
	IPR19	Initialized	Retained	Retained	—	Retained
	USDTENDRR	Initialized	Retained	Retained	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
UBC	BAR_0	Initialized	Retained	Retained	Retained	Retained
	BAMR_0	Initialized	Retained	Retained	Retained	Retained
	BBR_0	Initialized	Retained	Retained	Retained	Retained
	BAR_1	Initialized	Retained	Retained	Retained	Retained
	BAMR_1	Initialized	Retained	Retained	Retained	Retained
	BBR_1	Initialized	Retained	Retained	Retained	Retained
	BAR_2	Initialized	Retained	Retained	Retained	Retained
	BAMR_2	Initialized	Retained	Retained	Retained	Retained
	BBR_2	Initialized	Retained	Retained	Retained	Retained
	BAR_3	Initialized	Retained	Retained	Retained	Retained
	BAMR_3	Initialized	Retained	Retained	Retained	Retained
	BBR_3	Initialized	Retained	Retained	Retained	Retained
	BRCR	Initialized	Retained	Retained	Retained	Retained
DTC	DTCERA	Initialized	Retained	Retained	Retained	Retained
	DTCERB	Initialized	Retained	Retained	Retained	Retained
	DTCERC	Initialized	Retained	Retained	Retained	Retained
	DTCERD	Initialized	Retained	Retained	Retained	Retained
	DTCERE	Initialized	Retained	Retained	Retained	Retained
	DTCCR	Initialized	Retained	Retained	Retained	Retained
	DTCVBR	Initialized	Retained	Retained	Retained	Retained
BSC	CMNCR	Initialized	Retained	Retained	—	Retained
	CS0BCR	Initialized	Retained	Retained	—	Retained
	CS1BCR	Initialized	Retained	Retained	—	Retained
	CS2BCR	Initialized	Retained	Retained	—	Retained
	CS3BCR	Initialized	Retained	Retained	—	Retained
	CS4BCR	Initialized	Retained	Retained	—	Retained
	CS5BCR	Initialized	Retained	Retained	—	Retained
	CS6BCR	Initialized	Retained	Retained	—	Retained
	CS7BCR	Initialized	Retained	Retained	—	Retained
	CS0WCR	Initialized	Retained	Retained	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
BSC	CS1WCR	Initialized	Retained	Retained	—	Retained
	CS2WCR	Initialized	Retained	Retained	—	Retained
	CS3WCR	Initialized	Retained	Retained	—	Retained
	CS4WCR	Initialized	Retained	Retained	—	Retained
	CS5WCR	Initialized	Retained	Retained	—	Retained
	CS6WCR	Initialized	Retained	Retained	—	Retained
	CS7WCR	Initialized	Retained	Retained	—	Retained
	SDCR	Initialized	Retained	Retained	—	Retained
	RTCSR	Initialized	Retained (Flag processing continued)	Retained	—	Retained (Flag processing continued)
	RTCNT	Initialized	Retained (Count-up continued)	Retained	—	Retained (Count-up continued)
	RTCOR	Initialized	Retained	Retained	—	Retained
	BSCEHR	Initialized	Retained	Retained	—	Retained
DMAC	SAR_0	Initialized	Retained	Retained	Retained	Retained
	DAR_0	Initialized	Retained	Retained	Retained	Retained
	DMATCR_0	Initialized	Retained	Retained	Retained	Retained
	CHCR_0	Initialized	Retained	Retained	Retained	Retained
	RSAR_0	Initialized	Retained	Retained	Retained	Retained
	RDAR_0	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_0	Initialized	Retained	Retained	Retained	Retained
	SAR_1	Initialized	Retained	Retained	Retained	Retained
	DAR_1	Initialized	Retained	Retained	Retained	Retained
	DMATCR_1	Initialized	Retained	Retained	Retained	Retained
	CHCR_1	Initialized	Retained	Retained	Retained	Retained
	RSAR_1	Initialized	Retained	Retained	Retained	Retained
	RDAR_1	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_1	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
DMAC	SAR_2	Initialized	Retained	Retained	Retained	Retained
	DAR_2	Initialized	Retained	Retained	Retained	Retained
	DMATCR_2	Initialized	Retained	Retained	Retained	Retained
	CHCR_2	Initialized	Retained	Retained	Retained	Retained
	RSAR_2	Initialized	Retained	Retained	Retained	Retained
	RDAR_2	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_2	Initialized	Retained	Retained	Retained	Retained
	SAR_3	Initialized	Retained	Retained	Retained	Retained
	DAR_3	Initialized	Retained	Retained	Retained	Retained
	DMATCR_3	Initialized	Retained	Retained	Retained	Retained
	CHCR_3	Initialized	Retained	Retained	Retained	Retained
	RSAR_3	Initialized	Retained	Retained	Retained	Retained
	RDAR_3	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_3	Initialized	Retained	Retained	Retained	Retained
	SAR_4	Initialized	Retained	Retained	Retained	Retained
	DAR_4	Initialized	Retained	Retained	Retained	Retained
	DMATCR_4	Initialized	Retained	Retained	Retained	Retained
	CHCR_4	Initialized	Retained	Retained	Retained	Retained
	RSAR_4	Initialized	Retained	Retained	Retained	Retained
	RDAR_4	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_4	Initialized	Retained	Retained	Retained	Retained
	SAR_5	Initialized	Retained	Retained	Retained	Retained
	DAR_5	Initialized	Retained	Retained	Retained	Retained
	DMATCR_5	Initialized	Retained	Retained	Retained	Retained
	CHCR_5	Initialized	Retained	Retained	Retained	Retained
	RSAR_5	Initialized	Retained	Retained	Retained	Retained
	RDAR_5	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_5	Initialized	Retained	Retained	Retained	Retained
	SAR_6	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
DMAC	DAR_6	Initialized	Retained	Retained	Retained	Retained
	DMATCR_6	Initialized	Retained	Retained	Retained	Retained
	CHCR_6	Initialized	Retained	Retained	Retained	Retained
	RSAR_6	Initialized	Retained	Retained	Retained	Retained
	RDAR_6	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_6	Initialized	Retained	Retained	Retained	Retained
	SAR_7	Initialized	Retained	Retained	Retained	Retained
	DAR_7	Initialized	Retained	Retained	Retained	Retained
	DMATCR_7	Initialized	Retained	Retained	Retained	Retained
	CHCR_7	Initialized	Retained	Retained	Retained	Retained
	RSAR_7	Initialized	Retained	Retained	Retained	Retained
	RDAR_7	Initialized	Retained	Retained	Retained	Retained
	RDMATCR_7	Initialized	Retained	Retained	Retained	Retained
	DMAOR	Initialized	Retained	Retained	Retained	Retained
	DMARS0	Initialized	Retained	Retained	Retained	Retained
	DMARS1	Initialized	Retained	Retained	Retained	Retained
	DMARS2	Initialized	Retained	Retained	Retained	Retained
	DMARS3	Initialized	Retained	Retained	Retained	Retained
MTU2	TCR_0	Initialized	Retained	Retained	Initialized	Retained
	TMDR_0	Initialized	Retained	Retained	Initialized	Retained
	TIORH_0	Initialized	Retained	Retained	Initialized	Retained
	TIORL_0	Initialized	Retained	Retained	Initialized	Retained
	TIER_0	Initialized	Retained	Retained	Initialized	Retained
	TSR_0	Initialized	Retained	Retained	Initialized	Retained
	TCNT_0	Initialized	Retained	Retained	Initialized	Retained
	TGRA_0	Initialized	Retained	Retained	Initialized	Retained
	TGRB_0	Initialized	Retained	Retained	Initialized	Retained
	TGRC_0	Initialized	Retained	Retained	Initialized	Retained
	TGRD_0	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	TGRE_0	Initialized	Retained	Retained	Initialized	Retained
	TGRF_0	Initialized	Retained	Retained	Initialized	Retained
	TIER2_0	Initialized	Retained	Retained	Initialized	Retained
	TSR2_0	Initialized	Retained	Retained	Initialized	Retained
	TBTM_0	Initialized	Retained	Retained	Initialized	Retained
	TCR_1	Initialized	Retained	Retained	Initialized	Retained
	TMDR_1	Initialized	Retained	Retained	Initialized	Retained
	TIOR_1	Initialized	Retained	Retained	Initialized	Retained
	TIER_1	Initialized	Retained	Retained	Initialized	Retained
	TSR_1	Initialized	Retained	Retained	Initialized	Retained
	TCNT_1	Initialized	Retained	Retained	Initialized	Retained
	TGRA_1	Initialized	Retained	Retained	Initialized	Retained
	TGRB_1	Initialized	Retained	Retained	Initialized	Retained
	TICCR	Initialized	Retained	Retained	Initialized	Retained
	TCR_2	Initialized	Retained	Retained	Initialized	Retained
	TMDR_2	Initialized	Retained	Retained	Initialized	Retained
	TIOR_2	Initialized	Retained	Retained	Initialized	Retained
	TIER_2	Initialized	Retained	Retained	Initialized	Retained
	TSR_2	Initialized	Retained	Retained	Initialized	Retained
	TCNT_2	Initialized	Retained	Retained	Initialized	Retained
	TGRA_2	Initialized	Retained	Retained	Initialized	Retained
	TGRB_2	Initialized	Retained	Retained	Initialized	Retained
	TCR_3	Initialized	Retained	Retained	Initialized	Retained
	TMDR_3	Initialized	Retained	Retained	Initialized	Retained
	TIORH_3	Initialized	Retained	Retained	Initialized	Retained
	TIORL_3	Initialized	Retained	Retained	Initialized	Retained
	TIER_3	Initialized	Retained	Retained	Initialized	Retained
	TSR_3	Initialized	Retained	Retained	Initialized	Retained
	TCNT_3	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	TGRA_3	Initialized	Retained	Retained	Initialized	Retained
	TGRB_3	Initialized	Retained	Retained	Initialized	Retained
	TGRC_3	Initialized	Retained	Retained	Initialized	Retained
	TGRD_3	Initialized	Retained	Retained	Initialized	Retained
	TBTM_3	Initialized	Retained	Retained	Initialized	Retained
	TCR_4	Initialized	Retained	Retained	Initialized	Retained
	TMDR_4	Initialized	Retained	Retained	Initialized	Retained
	TIORH_4	Initialized	Retained	Retained	Initialized	Retained
	TIORL_4	Initialized	Retained	Retained	Initialized	Retained
	TIER_4	Initialized	Retained	Retained	Initialized	Retained
	TSR_4	Initialized	Retained	Retained	Initialized	Retained
	TCNT_4	Initialized	Retained	Retained	Initialized	Retained
	TGRA_4	Initialized	Retained	Retained	Initialized	Retained
	TGRB_4	Initialized	Retained	Retained	Initialized	Retained
	TGRC_4	Initialized	Retained	Retained	Initialized	Retained
	TGRD_4	Initialized	Retained	Retained	Initialized	Retained
	TBTM_4	Initialized	Retained	Retained	Initialized	Retained
	TADCR	Initialized	Retained	Retained	Initialized	Retained
	TADCORA_4	Initialized	Retained	Retained	Initialized	Retained
	TADCORB_4	Initialized	Retained	Retained	Initialized	Retained
	TADCOBRA_4	Initialized	Retained	Retained	Initialized	Retained
	TADCOBRB_4	Initialized	Retained	Retained	Initialized	Retained
	TCRU_5	Initialized	Retained	Retained	Initialized	Retained
	TCRV_5	Initialized	Retained	Retained	Initialized	Retained
	TCRW_5	Initialized	Retained	Retained	Initialized	Retained
	TIORU_5	Initialized	Retained	Retained	Initialized	Retained
	TIORV_5	Initialized	Retained	Retained	Initialized	Retained
	TIORW_5	Initialized	Retained	Retained	Initialized	Retained
	TIER_5	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	TSR_5	Initialized	Retained	Retained	Initialized	Retained
	TSTR_5	Initialized	Retained	Retained	Initialized	Retained
	TCNTU_5	Initialized	Retained	Retained	Initialized	Retained
	TCNTV_5	Initialized	Retained	Retained	Initialized	Retained
	TCNTW_5	Initialized	Retained	Retained	Initialized	Retained
	TGRU_5	Initialized	Retained	Retained	Initialized	Retained
	TGRV_5	Initialized	Retained	Retained	Initialized	Retained
	TGRW_5	Initialized	Retained	Retained	Initialized	Retained
	TCNTCMPCLR	Initialized	Retained	Retained	Initialized	Retained
	TSTR	Initialized	Retained	Retained	Initialized	Retained
	TSYR	Initialized	Retained	Retained	Initialized	Retained
	TCSYSTR	Initialized	Retained	Retained	Initialized	Retained
	TRWER	Initialized	Retained	Retained	Initialized	Retained
	TOER	Initialized	Retained	Retained	Initialized	Retained
	TOCR1	Initialized	Retained	Retained	Initialized	Retained
	TOCR2	Initialized	Retained	Retained	Initialized	Retained
	TGCR	Initialized	Retained	Retained	Initialized	Retained
	TCDR	Initialized	Retained	Retained	Initialized	Retained
	TDDR	Initialized	Retained	Retained	Initialized	Retained
	TCNTS	Initialized	Retained	Retained	Initialized	Retained
	TCBR	Initialized	Retained	Retained	Initialized	Retained
	TITCR	Initialized	Retained	Retained	Initialized	Retained
	TITCNT	Initialized	Retained	Retained	Initialized	Retained
	TBTER	Initialized	Retained	Retained	Initialized	Retained
	TDER	Initialized	Retained	Retained	Initialized	Retained
	TWCR	Initialized	Retained	Retained	Initialized	Retained
	TOLBR	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TCR_3S	Initialized	Retained	Retained	Initialized	Retained
	TMDR_3S	Initialized	Retained	Retained	Initialized	Retained
	TIORH_3S	Initialized	Retained	Retained	Initialized	Retained
	TIORL_3S	Initialized	Retained	Retained	Initialized	Retained
	TIER_3S	Initialized	Retained	Retained	Initialized	Retained
	TSR_3S	Initialized	Retained	Retained	Initialized	Retained
	TCNT_3S	Initialized	Retained	Retained	Initialized	Retained
	TGRA_3S	Initialized	Retained	Retained	Initialized	Retained
	TGRB_3S	Initialized	Retained	Retained	Initialized	Retained
	TGRC_3S	Initialized	Retained	Retained	Initialized	Retained
	TGRD_3S	Initialized	Retained	Retained	Initialized	Retained
	TBTM_3S	Initialized	Retained	Retained	Initialized	Retained
	TCR_4S	Initialized	Retained	Retained	Initialized	Retained
	TMDR_4S	Initialized	Retained	Retained	Initialized	Retained
	TIORH_4S	Initialized	Retained	Retained	Initialized	Retained
	TIORL_4S	Initialized	Retained	Retained	Initialized	Retained
	TIER_4S	Initialized	Retained	Retained	Initialized	Retained
	TSR_4S	Initialized	Retained	Retained	Initialized	Retained
	TCNT_4S	Initialized	Retained	Retained	Initialized	Retained
	TGRA_4S	Initialized	Retained	Retained	Initialized	Retained
	TGRB_4S	Initialized	Retained	Retained	Initialized	Retained
	TGRC_4S	Initialized	Retained	Retained	Initialized	Retained
	TGRD_4S	Initialized	Retained	Retained	Initialized	Retained
	TBTM_4S	Initialized	Retained	Retained	Initialized	Retained
	TADCRS	Initialized	Retained	Retained	Initialized	Retained
	TADCORA_4S	Initialized	Retained	Retained	Initialized	Retained
	TADCORB_4S	Initialized	Retained	Retained	Initialized	Retained
	TADCOBRA_4S	Initialized	Retained	Retained	Initialized	Retained
	TADCOBRB_4S	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TCRU_5S	Initialized	Retained	Retained	Initialized	Retained
	TCRV_5S	Initialized	Retained	Retained	Initialized	Retained
	TCRW_5S	Initialized	Retained	Retained	Initialized	Retained
	TIORU_5S	Initialized	Retained	Retained	Initialized	Retained
	TIORV_5S	Initialized	Retained	Retained	Initialized	Retained
	TIORW_5S	Initialized	Retained	Retained	Initialized	Retained
	TIER_5S	Initialized	Retained	Retained	Initialized	Retained
	TSR_5S	Initialized	Retained	Retained	Initialized	Retained
	TSTR_5S	Initialized	Retained	Retained	Initialized	Retained
	TCNTU_5S	Initialized	Retained	Retained	Initialized	Retained
	TCNTV_5S	Initialized	Retained	Retained	Initialized	Retained
	TCNTW_5S	Initialized	Retained	Retained	Initialized	Retained
	TGRU_5S	Initialized	Retained	Retained	Initialized	Retained
	TGRV_5S	Initialized	Retained	Retained	Initialized	Retained
	TGRW_5S	Initialized	Retained	Retained	Initialized	Retained
	TCNTCMPCLRS	Initialized	Retained	Retained	Initialized	Retained
	TSTRS	Initialized	Retained	Retained	Initialized	Retained
	TSYRS	Initialized	Retained	Retained	Initialized	Retained
	TRWERS	Initialized	Retained	Retained	Initialized	Retained
	TOERS	Initialized	Retained	Retained	Initialized	Retained
	TOCR1S	Initialized	Retained	Retained	Initialized	Retained
	TOCR2S	Initialized	Retained	Retained	Initialized	Retained
	TGCRS	Initialized	Retained	Retained	Initialized	Retained
	TCDRS	Initialized	Retained	Retained	Initialized	Retained
	TDDRS	Initialized	Retained	Retained	Initialized	Retained
	TCNTSS	Initialized	Retained	Retained	Initialized	Retained
	TCBRS	Initialized	Retained	Retained	Initialized	Retained
	TITCRS	Initialized	Retained	Retained	Initialized	Retained
	TITCNTS	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TBTERS	Initialized	Retained	Retained	Initialized	Retained
	TDERS	Initialized	Retained	Retained	Initialized	Retained
	TSYCRS	Initialized	Retained	Retained	Initialized	Retained
	TWCRS	Initialized	Retained	Retained	Initialized	Retained
	TOLBRS	Initialized	Retained	Retained	Initialized	Retained
POE2	ICSR1	Initialized	Retained	Retained	—	Retained
	OCSR1	Initialized	Retained	Retained	—	Retained
	ICSR2	Initialized	Retained	Retained	—	Retained
	OCSR2	Initialized	Retained	Retained	—	Retained
	ICSR3	Initialized	Retained	Retained	—	Retained
	SPOER	Initialized	Retained	Retained	—	Retained
	POECR1	Initialized	Retained	Retained	—	Retained
	POECR2	Initialized	Retained	Retained	—	Retained
CMT	CMSTR	Initialized	Retained	Retained	Initialized	Retained
	CMCSR_0	Initialized	Retained	Retained	Initialized	Retained
	CMCNT_0	Initialized	Retained	Retained	Initialized	Retained
	CMCOR_0	Initialized	Retained	Retained	Initialized	Retained
	CMCSR_1	Initialized	Retained	Retained	Initialized	Retained
	CMCNT_1	Initialized	Retained	Retained	Initialized	Retained
	CMCOR_1	Initialized	Retained	Retained	Initialized	Retained
WDT	WTCSR	Initialized	Retained*[4]	Initialized	—	Retained
	WTCNT	Initialized	Retained*[4]	Initialized	—	Retained
	WRCSR	Initialized*[1]	Retained	Initialized	—	Retained
SCI (channel 0)	SCSMR_0	Initialized	Retained	Retained	Initialized	Retained
	SCBRR_0	Initialized	Retained	Retained	Initialized	Retained
	SCSCR_0	Initialized	Retained	Retained	Initialized	Retained
	SCTDR_0	—	Retained	Retained	Initialized	Retained
	SCSSR_0	Initialized	Retained	Retained	Initialized	Retained
	SCRDR_0	—	Retained	Retained	Initialized	Retained
	SCSDCR_0	Initialized	Retained	Retained	Initialized	Retained
	SCSPTR_0	Initialized*[5]	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
SCI (channel 1)	SCSMR_1	Initialized	Retained	Retained	Initialized	Retained
	SCBRR_1	Initialized	Retained	Retained	Initialized	Retained
	SCSCR_1	Initialized	Retained	Retained	Initialized	Retained
	SCTDR_1	—	Retained	Retained	Initialized	Retained
	SCSSR_1	Initialized	Retained	Retained	Initialized	Retained
	SCRDR_1	—	Retained	Retained	Initialized	Retained
	SCSDCR_1	Initialized	Retained	Retained	Initialized	Retained
	SCSPTR_1	Initialized*[5]	Retained	Retained	Initialized	Retained
SCI (channel 2)	SCSMR_2	Initialized	Retained	Retained	Initialized	Retained
	SCBRR_2	Initialized	Retained	Retained	Initialized	Retained
	SCSCR_2	Initialized	Retained	Retained	Initialized	Retained
	SCTDR_2	—	Retained	Retained	Initialized	Retained
	SCSSR_2	Initialized	Retained	Retained	Initialized	Retained
	SCRDR_2	—	Retained	Retained	Initialized	Retained
	SCSDCR_2	Initialized	Retained	Retained	Initialized	Retained
	SCSPTR_2	Initialized*[5]	Retained	Retained	Initialized	Retained
SCI (channel 4)	SCSMR_4	Initialized	Retained	Retained	Initialized	Retained
	SCBRR_4	Initialized	Retained	Retained	Initialized	Retained
	SCSCR_4	Initialized	Retained	Retained	Initialized	Retained
	SCTDR_4	—	Retained	Retained	Initialized	Retained
	SCSSR_4	Initialized	Retained	Retained	Initialized	Retained
	SCRDR_4	—	Retained	Retained	Initialized	Retained
	SCSDCR_4	Initialized	Retained	Retained	Initialized	Retained
	SCSPTR_4	Initialized*[5]	Retained	Retained	Initialized	Retained
SCIF	SCSMR_3	Initialized	Retained	Retained	Retained	Retained
	SCBRR_3	Initialized	Retained	Retained	Retained	Retained
	SCSCR_3	Initialized	Retained	Retained	Retained	Retained
	SCFTDR_3	—	Retained	Retained	Retained	Retained
	SCFSR_3	Initialized	Retained	Retained	Retained	Retained
	SCFRDR_3	—	Retained	Retained	Retained	Retained
	SCFCR_3	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
SCIF	SCFDR_3	Initialized	Retained	Retained	Retained	Retained
	SCSPTR_3	Initialized*[5]	Retained	Retained	Retained	Retained
	SCLSR_3	Initialized	Retained	Retained	Retained	Retained
	SCSEMR_3	Initialized	Retained	Retained	Retained	Retained
RSPI	SPCR	Initialized	Retained	Retained	Initialized	Retained
	SSLP	Initialized	Retained	Retained	Initialized	Retained
	SPPCR	Initialized	Retained	Retained	Initialized	Retained
	SPSR	Initialized	Retained	Retained	Initialized	Retained
	SPDR	Initialized	Retained	Retained	Initialized	Retained
	SPSCR	Initialized	Retained	Retained	Initialized	Retained
	SPSSR	Initialized	Retained	Retained	Initialized	Retained
	SPBR	Initialized	Retained	Retained	Initialized	Retained
	SPDCR	Initialized	Retained	Retained	Initialized	Retained
	SPCKD	Initialized	Retained	Retained	Initialized	Retained
	SSLND	Initialized	Retained	Retained	Initialized	Retained
	SPND	Initialized	Retained	Retained	Initialized	Retained
	SPCMD0	Initialized	Retained	Retained	Initialized	Retained
	SPCMD1	Initialized	Retained	Retained	Initialized	Retained
	SPCMD2	Initialized	Retained	Retained	Initialized	Retained
	SPCMD3	Initialized	Retained	Retained	Initialized	Retained
IIC3	ICCR1	Initialized	Retained	Retained	Retained	Retained
	ICCR2	Initialized	Retained	Retained	Retained	Retained
	ICMR	Initialized	Retained	Retained/ Initialized (bc2-0)	Retained/ Initialized (bc2-0)	Retained
	ICIER	Initialized	Retained	Retained	Retained	Retained
	ICSR	Initialized	Retained	Retained	Retained	Retained
	SAR	Initialized	Retained	Retained	Retained	Retained
	ICDRT	Initialized	Retained	Retained	Retained	Retained
	ICDRR	Initialized	Retained	Retained	Retained	Retained
	NF2CYC	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
ADC	ADCR_0	Initialized	Retained	Initialized	Retained	Retained
	ADSR_0	Initialized	Retained	Initialized	Retained	Retained
	ADSTRGR_0	Initialized	Retained	Initialized	Retained	Retained
	ADANSR_0	Initialized	Retained	Initialized	Retained	Retained
	ADBYPSCR_0	Initialized	Retained	Initialized	Retained	Retained
	ADDR0	Initialized	Retained	Initialized	Retained	Retained
	ADDR1	Initialized	Retained	Initialized	Retained	Retained
	ADDR2	Initialized	Retained	Initialized	Retained	Retained
	ADDR3	Initialized	Retained	Initialized	Retained	Retained
	ADCR_1	Initialized	Retained	Initialized	Retained	Retained
	ADSR_1	Initialized	Retained	Initialized	Retained	Retained
	ADSTRGR_1	Initialized	Retained	Initialized	Retained	Retained
	ADANSR_1	Initialized	Retained	Initialized	Retained	Retained
	ADBYPSCR_1	Initialized	Retained	Initialized	Retained	Retained
	ADDR4	Initialized	Retained	Initialized	Retained	Retained
	ADDR5	Initialized	Retained	Initialized	Retained	Retained
	ADDR6	Initialized	Retained	Initialized	Retained	Retained
	ADDR7	Initialized	Retained	Initialized	Retained	Retained
RCAN-ET	MCR	Initialized	Retained	Retained	Initialized	Retained
	GSR	Initialized	Retained	Retained	Initialized	Retained
	BCR1	Initialized	Retained	Retained	Initialized	Retained
	BCR0	Initialized	Retained	Retained	Initialized	Retained
	IRR	Initialized	Retained	Retained	Initialized	Retained
	IMR	Initialized	Retained	Retained	Initialized	Retained
	TEC/REC	Initialized	Retained	Retained	Initialized	Retained
	TXPR1, 0	Initialized	Retained	Retained	Initialized	Retained
	TXCR0	Initialized	Retained	Retained	Initialized	Retained
	TXACK0	Initialized	Retained	Retained	Initialized	Retained
	ABACK0	Initialized	Retained	Retained	Initialized	Retained
	RXPR0	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
RCAN-ET	RFPR0	Initialized	Retained	Retained	Initialized	Retained
	MBIMR0	Initialized	Retained	Retained	Initialized	Retained
	UMSR0	Initialized	Retained	Retained	Initialized	Retained
	MB[0]. CONTROL0H	—	Retained	—	—	Retained
	MB[0]. CONTROL0L	—	Retained	—	—	Retained
	MB[0]. LAFMH	—	Retained	—	—	Retained
	MB[0]. LAFML	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[0]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[1]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[2]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[3]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[4]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[5]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[6]	—	Retained	—	—	Retained
	MB[0]. MSG_DATA[7]	—	Retained	—	—	Retained
	MB[0]. CONTROL1H	Initialized	Retained	Retained	Retained	Retained
	MB[0]. CONTROL1L	Initialized	Retained	Retained	Retained	Retained
	MB[1].	Same as MB[0]				
	MB[2].	Same as MB[0]				

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
RCAN-ET	MB[3].	Same as MB[0]				
	↓	(Ditto)				
	MB[13].	Same as MB[0]				
	MB[14].	Same as MB[0]				
	MB[15].	Same as MB[0]				
PFC	PAIORH	Initialized	Retained	Retained	—	Retained
	PAIORL	Initialized	Retained	Retained	—	Retained
	PACRH2	Initialized	Retained	Retained	—	Retained
	PACRH1	Initialized	Retained	Retained	—	Retained
	PACRL4	Initialized	Retained	Retained	—	Retained
	PACRL3	Initialized	Retained	Retained	—	Retained
	PACRL2	Initialized	Retained	Retained	—	Retained
	PACRL1	Initialized	Retained	Retained	—	Retained
	PAPCRH	Initialized	Retained	Retained	—	Retained
	PAPCRL	Initialized	Retained	Retained	—	Retained
	PBIORL	Initialized	Retained	Retained	—	Retained
	PBCRL4	Initialized	Retained	Retained	—	Retained
	PBCRL3	Initialized	Retained	Retained	—	Retained
	PBCRL2	Initialized	Retained	Retained	—	Retained
	PBCRL1	Initialized	Retained	Retained	—	Retained
	PBPCRL	Initialized	Retained	Retained	—	Retained
	PCIORL	Initialized	Retained	Retained	—	Retained
	PCCRL4	Initialized	Retained	Retained	—	Retained
	PCCRL3	Initialized	Retained	Retained	—	Retained
	PCCRL2	Initialized	Retained	Retained	—	Retained
	PCCRL1	Initialized	Retained	Retained	—	Retained
	PCPCRL	Initialized	Retained	Retained	—	Retained
	PDIORH	Initialized	Retained	Retained	—	Retained
	PDIORL	Initialized	Retained	Retained	—	Retained
	PDCRH4	Initialized	Retained	Retained	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
PFC	PDCRH3	Initialized	Retained	Retained	—	Retained
	PDCRH2	Initialized	Retained	Retained	—	Retained
	PDCRH1	Initialized	Retained	Retained	—	Retained
	PDCRL4	Initialized	Retained	Retained	—	Retained
	PDCRL3	Initialized	Retained	Retained	—	Retained
	PDCRL2	Initialized	Retained	Retained	—	Retained
	PDCRL1	Initialized	Retained	Retained	—	Retained
	PDPCRH	Initialized	Retained	Retained	—	Retained
	PDPCRL	Initialized	Retained	Retained	—	Retained
	PEIORL	Initialized	Retained	Retained	—	Retained
	PECRL4	Initialized	Retained	Retained	—	Retained
	PECRL3	Initialized	Retained	Retained	—	Retained
	PECRL2	Initialized	Retained	Retained	—	Retained
	PECRL1	Initialized	Retained	Retained	—	Retained
	HCPCR	Initialized	Retained	Retained	—	Retained
	IFCR	Initialized	Retained	Retained	—	Retained
	PEPCRL	Initialized	Retained	Retained	—	Retained
	PDACKCR	Initialized	Retained	Retained	—	Retained
I/O port	PADRH	Initialized	Retained	Retained	—	Retained
	PADRL	Initialized	Retained	Retained	—	Retained
	PAPRH	—	Retained	Retained	—	Retained
	PAPRL	—	Retained	Retained	—	Retained
	PBDRL	Initialized	Retained	Retained	—	Retained
	PBPRL	—	Retained	Retained	—	Retained
	PCDRL	Initialized	Retained	Retained	—	Retained
	PCPRL	—	Retained	Retained	—	Retained
	PDDRH	Initialized	Retained	Retained	—	Retained
	PDDRL	Initialized	Retained	Retained	—	Retained
	PDPRH	—	Retained	Retained	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
I/O port	PDPRL	—	Retained	Retained	—	Retained
	PEDRL	Initialized	Retained	Retained	—	Retained
	PEPRL	—	Retained	Retained	—	Retained
	PFDRL	—	Retained	Retained	—	Retained
USB	USBIFR0	Initialized*5	Retained	Retained	Retained	Retained
	USBIFR1	Initialized	Retained	Retained	Retained	Retained
	USBIFR2	Initialized	Retained	Retained	Retained	Retained
	USBIFR3	Initialized	Retained	Retained	Retained	Retained
	USBIFR4	Initialized	Retained	Retained	Retained	Retained
	USBIER0	Initialized	Retained	Retained	Retained	Retained
	USBIER1	Initialized	Retained	Retained	Retained	Retained
	USBIER2	Initialized	Retained	Retained	Retained	Retained
	USBIER3	Initialized	Retained	Retained	Retained	Retained
	USBIER4	Initialized	Retained	Retained	Retained	Retained
	USBISR0	Initialized	Retained	Retained	Retained	Retained
	USBISR1	Initialized	Retained	Retained	Retained	Retained
	USBISR2	Initialized	Retained	Retained	Retained	Retained
	USBISR3	Initialized	Retained	Retained	Retained	Retained
	USBISR4	Initialized	Retained	Retained	Retained	Retained
	USBEPDR0i	—	Retained	Retained	Retained	Retained
	USBEPDR0o	—	Retained	Retained	Retained	Retained
	USBEPDR0s	—	Retained	Retained	Retained	Retained
	USBEPDR1	—	Retained	Retained	Retained	Retained
	USBEPDR2	—	Retained	Retained	Retained	Retained
	USBEPDR3	—	Retained	Retained	Retained	Retained
	USBEPDR4	—	Retained	Retained	Retained	Retained
	USBEPDR5	—	Retained	Retained	Retained	Retained
	USBEPDR6	—	Retained	Retained	Retained	Retained
	USBEPDR7	—	Retained	Retained	Retained	Retained
	USBEPDR8	—	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
USB	USBEPDR9	—	Retained	Retained	Retained	Retained
	USBEPSZ0o	Initialized	Retained	Retained	Retained	Retained
	USBEPSZ1	Initialized	Retained	Retained	Retained	Retained
	USBEPSZ4	Initialized	Retained	Retained	Retained	Retained
	USBEPSZ7	Initialized	Retained	Retained	Retained	Retained
	USBDASTS0	Initialized	Retained	Retained	Retained	Retained
	USBDASTS1	Initialized	Retained	Retained	Retained	Retained
	USBDASTS2	Initialized	Retained	Retained	Retained	Retained
	USBDASTS3	Initialized	Retained	Retained	Retained	Retained
	USBTRG0	Initialized	Retained	Retained	Retained	Retained
	USBTRG1	Initialized	Retained	Retained	Retained	Retained
	USBTRG2	Initialized	Retained	Retained	Retained	Retained
	USBTRG3	Initialized	Retained	Retained	Retained	Retained
	USBFCLR0	Initialized	Retained	Retained	Retained	Retained
	USBFCLR1	Initialized	Retained	Retained	Retained	Retained
	USBFCLR2	Initialized	Retained	Retained	Retained	Retained
	USBFCLR3	Initialized	Retained	Retained	Retained	Retained
	USBEPSTL0	Initialized	Retained	Retained	Retained	Retained
	USBEPSTL1	Initialized	Retained	Retained	Retained	Retained
	USBEPSTL2	Initialized	Retained	Retained	Retained	Retained
	USBEPSTL3	Initialized	Retained	Retained	Retained	Retained
	USBSTLSR1	Initialized	Retained	Retained	Retained	Retained
	USBSTLSR2	Initialized	Retained	Retained	Retained	Retained
	USBSTLSR3	Initialized	Retained	Retained	Retained	Retained
	USBDMAR	Initialized	Retained	Retained	Retained	Retained
	USBCVR	Initialized	Retained	Retained	Retained	Retained
	USBCTLR	Initialized	Retained	Retained	Retained	Retained
	USBEPIR	—	Retained	Retained	Retained	Retained
	USBTRNTREG0	Initialized	Retained	Retained	Retained	Retained
	USBTRNTREG1	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
E-DMAC	EDMR	Initialized	Retained	Retained	Retained	Retained
	EDTRR	Initialized	Retained	Retained	Retained	Retained
	EDRRR	Initialized	Retained	Retained	Retained	Retained
	TDLAR	Initialized	Retained	Retained	Retained	Retained
	RDLAR	Initialized	Retained	Retained	Retained	Retained
	EESR	Initialized	Retained	Retained	Retained	Retained
	EESIPR	Initialized	Retained	Retained	Retained	Retained
	TRSCER	Initialized	Retained	Retained	Retained	Retained
	RMFCR	Initialized	Retained	Retained	Retained	Retained
	TFTR	Initialized	Retained	Retained	Retained	Retained
	FDR	Initialized	Retained	Retained	Retained	Retained
	RMCR	Initialized	Retained	Retained	Retained	Retained
	TFUCR	Initialized	Retained	Retained	Retained	Retained
	RFOCR	Initialized	Retained	Retained	Retained	Retained
	IOSR	Initialized	Retained	Retained	Retained	Retained
	EDOCR	Initialized	Retained	Retained	Retained	Retained
	FCFTR	Initialized	Retained	Retained	Retained	Retained
	TRIMD	Initialized	Retained	Retained	Retained	Retained
	RBWAR	Initialized	Retained	Retained	Retained	Retained
	RDFAR	Initialized	Retained	Retained	Retained	Retained
	TBRAR	Initialized	Retained	Retained	Retained	Retained
	TDFAR	Initialized	Retained	Retained	Retained	Retained
EtherC	ECMR	Initialized	Retained	Retained	Retained	Retained
	ECSR	Initialized	Retained	Retained	Retained	Retained
	ECSIPR	Initialized	Retained	Retained	Retained	Retained
	PIR	Initialized*5	Retained	Retained	Retained	Retained
	MAHR	Initialized	Retained	Retained	Retained	Retained
	MALR	Initialized	Retained	Retained	Retained	Retained
	RFLR	Initialized	Retained	Retained	Retained	Retained
	PSR	Initialized*5	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
EtherC	TROCR	Initialized	Retained	Retained	Retained	Retained
	CDCR	Initialized	Retained	Retained	Retained	Retained
	LCCR	Initialized	Retained	Retained	Retained	Retained
	CNDCR	Initialized	Retained	Retained	Retained	Retained
	CEFCR	Initialized	Retained	Retained	Retained	Retained
	FRECR	Initialized	Retained	Retained	Retained	Retained
	TSFRCR	Initialized	Retained	Retained	Retained	Retained
	TLFRCR	Initialized	Retained	Retained	Retained	Retained
	RFCR	Initialized	Retained	Retained	Retained	Retained
	MAFCR	Initialized	Retained	Retained	Retained	Retained
	IPGR	Initialized	Retained	Retained	Retained	Retained
	APR	Initialized	Retained	Retained	Retained	Retained
	MPR	Initialized	Retained	Retained	Retained	Retained
	TPAUSER	Initialized	Retained	Retained	Retained	Retained
	RDMLR	Initialized	Retained	Retained	Retained	Retained
	RFCF	Initialized	Retained	Retained	Retained	Retained
	TPAUSECR	Initialized	Retained	Retained	Retained	Retained
	BCFRR	Initialized	Retained	Retained	Retained	Retained
ROM/FLD	FPMON	Initialized	Retained	Retained	Retained	Retained
	FMODR	Initialized	Retained	Retained	Retained	Retained
	FASTAT	Initialized	Retained	Retained	Retained	Retained
	FAEINT	Initialized	Retained	Retained	Retained	Retained
	ROMMAT	Initialized	Retained	Retained	Retained	Retained
	FCURAME	Initialized	Retained	Retained	Retained	Retained
	FSTATR0	Initialized	Retained	Retained	Retained	Retained
	FSTATR1	Initialized	Retained	Retained	Retained	Retained
	FENTRYR	Initialized	Retained	Retained	Retained	Retained
	FPROTR	Initialized	Retained	Retained	Retained	Retained
	FRESETR	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
ROM/FLD	FCMDR	Initialized	Retained	Retained	Retained	Retained
	FCPSR	Initialized	Retained	Retained	Retained	Retained
	EEPBCCNT	Initialized	Retained	Retained	Retained	Retained
	FPESTAT	Initialized	Retained	Retained	Retained	Retained
	EEPBCSTAT	Initialized	Retained	Retained	Retained	Retained
	EEPRE0	Initialized	Retained	Retained	Retained	Retained
	EEPWE0	Initialized	Retained	Retained	Retained	Retained
	RCCR	Initialized	Retained	Retained	Retained	Retained
	PCKAR	Initialized	Retained	Retained	Retained	Retained
Power-down mode	STBCR	Initialized	Retained	Retained	—	Retained
	STBCR2	Initialized	Retained	Retained	—	Retained
	SYSCR1	Initialized	Retained	Retained	—	Retained
	SYSCR2	Initialized	Retained	Retained	—	Retained
	STBCR3	Initialized	Retained	Retained	—	Retained
	STBCR4	Initialized	Retained	Retained	—	Retained
	STBCR5	Initialized	Retained	Retained	—	Retained
	STBCR6	Initialized	Retained	Retained	—	Retained
H-UDI*3	SDIR	Retained	Retained	Retained	Retained	Retained

Notes: 1. Retains the previous value after an internal power-on reset by means of the WDT.
2. Bits BN[3:0] are initialized.
3. Initialized by TRST assertion or in the Test-Logic-Reset state of the TAP controller.
4. Initialized after an internal manual reset by means of the WDT.
5. Some bits are not initialized.

Section 33 Electrical Characteristics

33.1 Absolute Maximum Ratings

Table 33.1 lists the absolute maximum ratings.

Table 33.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (Internal)		$V_{CC}Q$, $PLLV_{CC}$, DrV_{CC}	–0.3 to +4.6	V
Input voltage (except analog input pins)		Vin	–0.3 to $V_{CC}Q$ +0.3	V
Analog power supply voltage		AV_{CC}	–0.3 to +7.0	V
Analog reference voltage		AVREF	–0.3 to AV_{CC} +0.3	V
Analog input voltage		V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	Industrial specifications	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. Supply the DrV_{CC} with the same voltage as the $V_{CC}Q$.

33.2 DC Characteristics

Tables 33.2 and 33.3 list DC characteristics.

Table 33.2 DC Characteristics (1) [Common Items]

Conditions: Ta = –40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		$V_{CC}Q$, $PLLV_{CC}$, DrV_{CC}*3	3.0	3.3	3.6	V	
Analog power supply voltage		AV_{CC}	4.5	5.0	5.5	V	
Supply current*1	Normal operation	I_{CC}	—	150	200	mA	Iϕ = 200 MHz Bϕ = 50 MHz Pϕ = 50 MHz
			—	85	120	mA	Iϕ = 100 MHz Bϕ = 50 MHz Pϕ = 50 MHz
	Software standby mode	I_{stby}	—	30	70	mA	VccQ = 3.3 V
	Sleep mode	I_{sleep}	—	100	140	mA	Iϕ = 200 MHz Bϕ = 50 MHz Pϕ = 50 MHz
			—	80	110	mA	Iϕ = 100 MHz Bϕ = 50 MHz Pϕ = 50 MHz
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Three-state leakage current	Input/output pins, all output pins (off state)	$\|I_{STI}\|$	—	—	1	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
Input capacitance	All pins	C_{in}	—	—	20	pF	
Analog power supply current	During A/D conversion	AI_{CC}	—	3	4	mA	Per 1 module
	Waiting for A/D conversion		—	30	50	mA	Per 1 module

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Reference power supply current	During A/D conversion	Alref	—	1	2	mA	Per 1 module
	Waiting for A/D conversion		—	0.8	1	mA	Per 1 module

Caution: When the A/D converter is not in use, the AV_{CC} and AV_{SS} pins should not be open. Connect the AV_{CC} to the $V_{CC}Q$.

Notes: 1. Supply current values are when all output and pull-up pins are unloaded.
2. I_{CC}, I_{sleep}, and I_{stby} represent the total currents consumed in the $V_{CC}Q$ and $PLLV_{CC}$ systems.
3. Be sure to supply the DrV_{CC} with the same voltage as the $V_{CC}Q$.

Table 33.2 DC Characteristics (2) [Except for I²C-Related Pins]

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5 V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFV_{SS} = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD1, MD0, FWE, $\overline{ASEMD0}$, $\overline{TRST}$, EXTAL, USBEXTAL	V_{IH}	$V_{CC}Q - 0.5$	—	$V_{CC}Q + 0.3$	V	$V_{CC}Q = 3.0$ to 3.6 V
	Analog ports		2.2	—	$AV_{CC} + 0.3$	V	$AV_{CC} = 3.0$ to 5.5 V*
	Input pins other than above (excluding Schmitt pins)		2.2	—	$V_{CC}Q + 0.3$		$V_{CC}Q = 3.0$ to 3.6 V
Input low voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD1, MD0, FWE, $\overline{ASEMD0}$, $\overline{TRST}$, EXTAL, USBXTAL	V_{IL}	–0.3	—	0.5	V	$V_{CC}Q = 3.0$ to 3.6 V
	Input pins other than above (excluding Schmitt pins)		–0.3	—	0.8	V	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input characteristics	TIOC0A to TIOC0D, TIOC1A, TIOC1B, TIOC2A, TIOC2B, TIOC3A to TIOC3D, TIOC4A to TIOC4D, TIC5U to TIC5W, TCLKA to TCLKD, TIOC3AS to TIOC3DS, TIOC4AS to TIOC4DS, TIC5US, TIC5VS, TIC5WS, POE8 and POE4 to POE0, SCK4 to SCK0, RxD4 to RxD0, IRQ7 to IRQ0, SCL, SDA, RSPCK, AMOSI, AMISO, ASSL0	V_T^+	$V_{CC}Q - 0.5$	—	—	V	$V_{CC}Q$ = 3.0 to 3.6 V
		V_T^-	—	—	0.5	V	
		$V_T^+ - V_T^-$	$V_{CC}Q \times 0.05$	—	—	V	
Output high voltage	All output pins	V_{OH}	$V_{CC}Q - 0.5$	—	—	V	I_{OH} = –200 μA
	TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS		$V_{CC}Q - 1.0$	—	—	V	I_{OH} = –5 mA
Output low voltage	TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS	V_{OL}	—	—	0.9	V	I_{OL} = 10 mA, $V_{CC}Q$ = 3.0 to 3.6 V
	SCL, SDA		—	—	0.4		I_{OL} = 3 mA
			—	—	0.5		I_{OL} = 8 mA
	All output pins except for above pins		—	—	0.4		I_{OL} = 1.6 mA
Input pull-up MOS current	Ports A, B, C, D, and E $\overline{\text{ASEMD0}}$	$-I_P$	–10	—	–800	μA	Vin = 0 V
RAM standby voltage		V_{RAM}	2.7	—	—	V	$V_{CC}Q$

Note: * When the A/D converter is in use, AV_{CC} must be from 4.5 to 5.5 V. When it is not in use, connect the AV_{CC} to the $V_{CC}Q$.

Table 33.3 Permissible Output Currents

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2.0*1*2	mA
Permissible output low current (total)	ΣI_{OL}	—	—	80	mA
Permissible output high current (per pin)	$-I_{OH}$	—	—	2	mA
Permissible output high current (total)	$\Sigma -I_{OH}$	—	—	25	mA

Notes: 1. TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS: I_{OL} = 15mA (Max)/-I_{OH} = 5mA.
SCL and SDA: I_{OL} = 8 mA (Max).
Of these pins, the number of pins from which current more than 2.0 mA runs evenly should be 3 or less.
2. Pins except USD+, USD-

Caution: To protect the LSI's reliability, do not exceed the output current values in table 33.3.

33.3 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed.

Table 33.4 Maximum Operating Frequency

Conditions: $V_{cc}Q = PLLV_{cc} = DrV_{cc} = 3.0$ to 3.6 V, $AV_{cc} = AVREF = 4.5$ to 5.5 V,
$V_{ss} = PLLV_{ss} = DrV_{ss} = AVREFVSS = AV_{ss} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

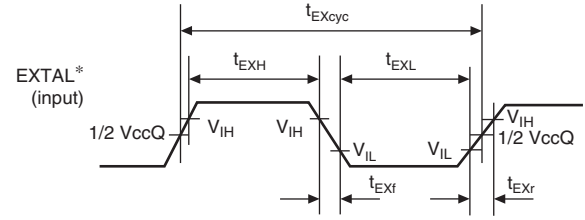
Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU (Iϕ)	f	20	—	200	MHz	
	Internal bus, external bus (Bϕ)		20	—	50		
	Peripheral module (Pϕ)		20	—	50		
	MTU2S (Mϕ)		40	—	100		
	AD (Aϕ)		40	—	50		

33.3.1 Clock Timing

Table 33.5 Clock Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5 V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = −40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	12.5	MHz	Figure 33.1
EXTAL clock input cycle time	t_{EXcyc}	80	100	ns	
EXTAL clock input pulse low width	t_{EXL}	20	—	ns	
EXTAL clock input pulse high width	t_{EXH}	20	—	ns	
EXTAL clock input rise time	t_{EXr}	—	5	ns	
EXTAL clock input fall time	t_{EXf}	—	5	ns	
CK clock output frequency	f_{OP}	20	50	MHz	Figure 33.2
CK clock output cycle time	t_{cyc}	20	50	ns	
CK clock output pulse low width	t_{CKOL}	4	—	ns	
CK clock output pulse high width	t_{CKOH}	4	—	ns	
CK clock output rise time	t_{CKOr}	—	3	ns	
CK clock output fall time	t_{CKOf}	—	3	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	Figure 33.3
Oscillation settling time on return from standby 1	t_{OSC2}	10	—	ms	Figure 33.4
Oscillation settling time on return from standby 2	t_{OSC3}	10	—	ms	Figure 33.5
USB clock power-on oscillation setting time	t_{OSC4}	8	—	ms	Figure 33.3
USB clock input frequency	f_{USB}		48	MHz	—
USB clock input cycle time	f_{USBcyc}		20.8	ns	—



Note: * When the clock is input on the EXTAL pin.

Figure 33.1 EXTAL Clock Input Timing

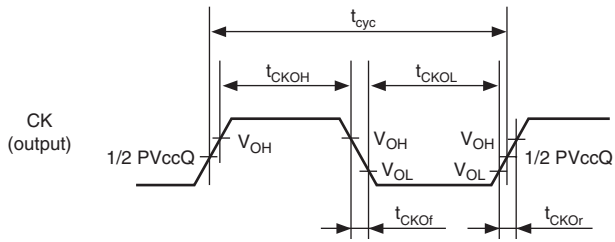
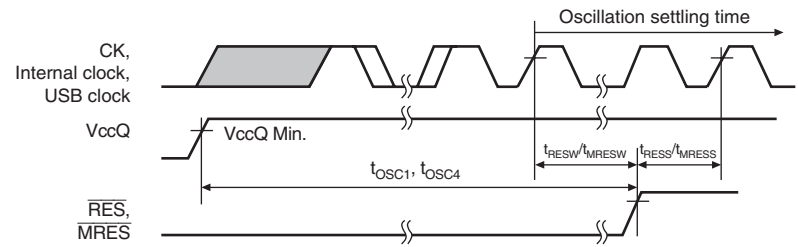
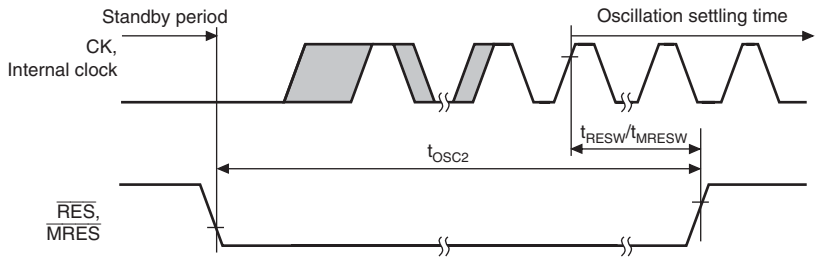


Figure 33.2 CK Clock Output Timing



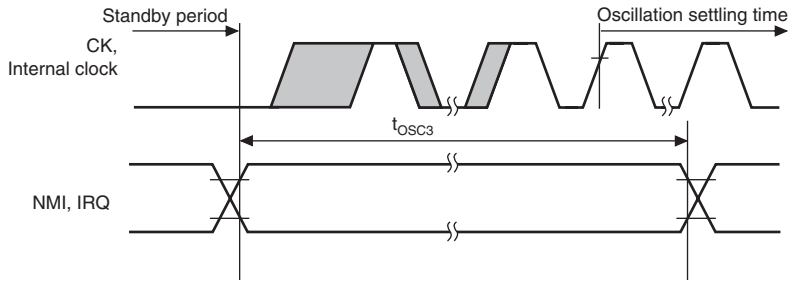
Note: Oscillation settling time when the internal oscillator is used.

Figure 33.3 Power-On Oscillation Settling Time



Note: Oscillation settling time when the internal oscillator is used.

Figure 33.4 Oscillation Settling Time on Return from Standby (Return by Reset)



Note: Oscillation settling time when the internal oscillator is used.

Figure 33.5 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

33.3.2 Control Signal Timing

Table 33.6 Control Signal Timing

Conditions: $V_{cc}Q = PLLV_{cc} = DrV_{cc} = 3.0$ to 3.6 V, $AV_{cc} = AVREF = 4.5$ to 5.5 V,
$V_{ss} = PLLV_{ss} = DrV_{ss} = AVREFVSS = AV_{ss} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

		Bϕ = 50 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{RES}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*[4]	—	t_{cyc}	Figures 33.3 to 33.6
		1.5*[4]	—	µs	
$\overline{RES}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100	—	µs	
$\overline{RES}$ setup time*[1]	t_{RESS}	65	—	ns	
$\overline{RES}$ hold time	t_{RESH}	15	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20*[3]	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	100	—	ns	
$\overline{MRES}$ hold time	t_{MRESH}	15	—	ns	
MD1, MD0, FWE setup time	t_{MDS}	20	—	t_{cyc}	Figure 33.6
$\overline{BREQ}$ setup time	t_{BREQS}	$1/2t_{cyc}$ + 15	—	ns	Figure 33.8
$\overline{BREQ}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 10	—	ns	
NMI setup time*[1]	t_{NMIS}	60	—	ns	Figure 33.7
NMI hold time	t_{NMIH}	10	—	ns	
IRQ7 to IRQ0 setup time*[1]	t_{IRQS}	35	—	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	10	—	ns	
IRQ pulse width	t_{IRQW}	4*[4]	—	t_{cyc}	
NMI pulse width	t_{NMIW}	4*[4]	—	t_{cyc}	
$\overline{IRQOUT}$/$\overline{REFOUT}$ output delay time	t_{IRQOD}	—	100	ns	Figure 33.9
$\overline{BACK}$ delay time	t_{BACKD}	—	$1/2t_{cyc}$ + 20	ns	Figure 33.8
Bus tri-state delay time 1	t_{BOFF1}	0	100	ns	
Bus tri-state delay time 2	t_{BOFF2}	0	100	ns	
Bus buffer on time 1	t_{BON1}	0	30	ns	
Bus buffer on time 2	t_{BON2}	0	30	ns	

Notes: 1. $\overline{RES}$, NMI, and IRQ7 to IRQ0 are asynchronous signals. When these setup times are observed, a change of these signals is detected at the clock rising edge. If the setup times are not observed, detection of a signal change may be delayed until the next rising edge of the clock.

2. In standby mode or when the clock multiplication ratio is changed, $t_{RESW} = t_{OSC2}$ (10 ms). Since the CK width is initialized by the $\overline{RES}$ pin, t_{cyc} becomes the initial value.
3. In standby mode, $t_{MRESW} = t_{OSC2}$ (10 ms).
4. Input the reset pulse over t_{RESW1} so that all conditions are met.

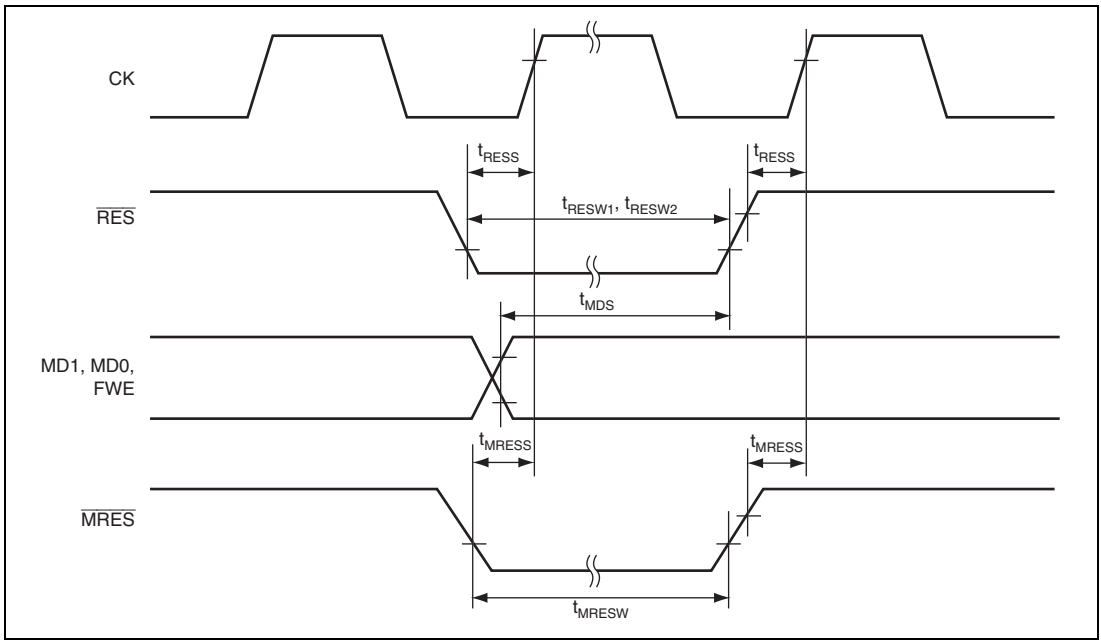


Figure 33.6 Reset Input Timing

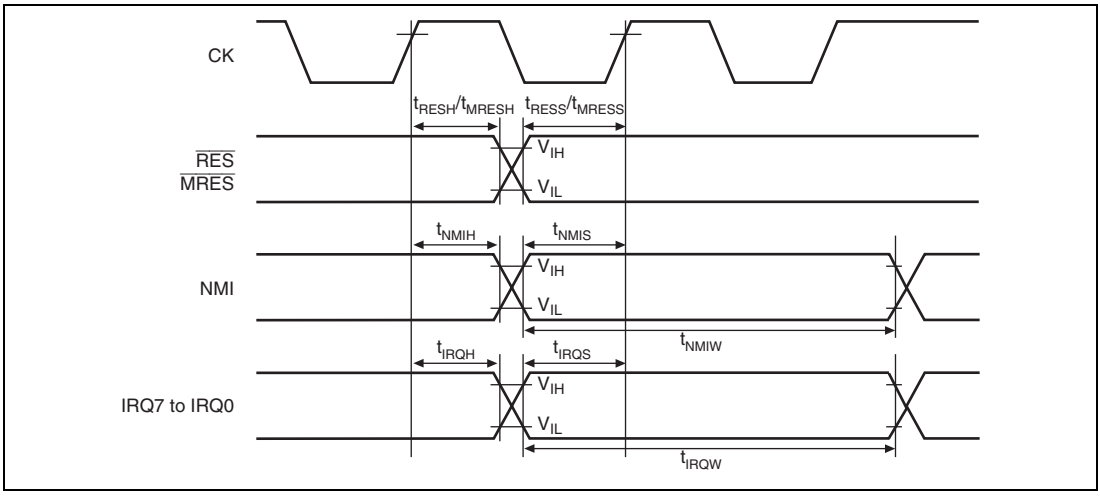


Figure 33.7 Interrupt Signal Input Timing

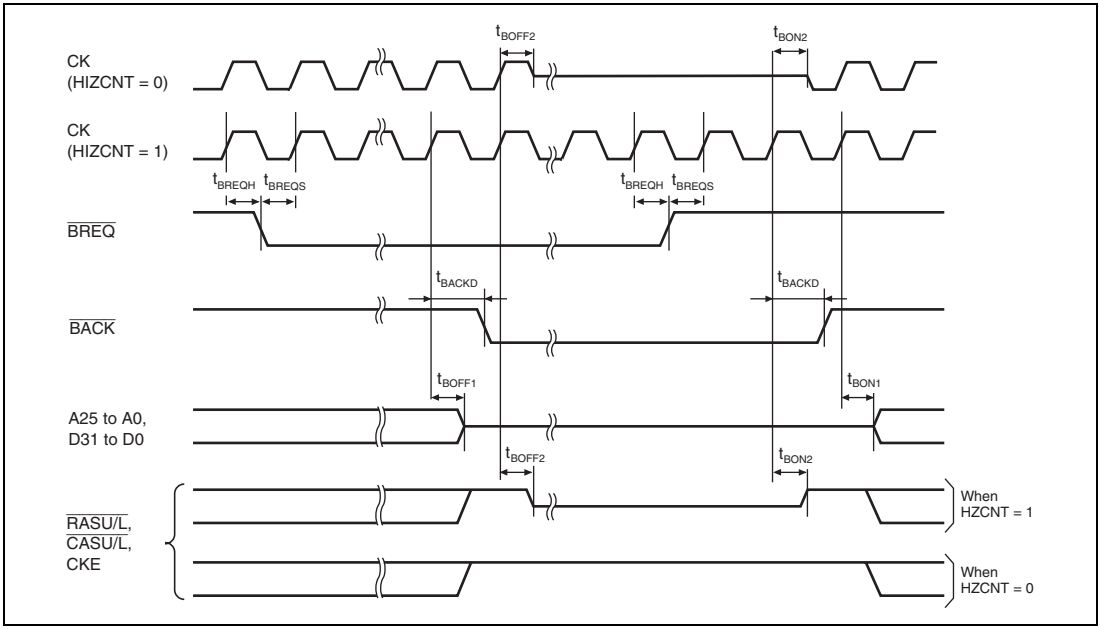


Figure 33.8 Bus Release Timing

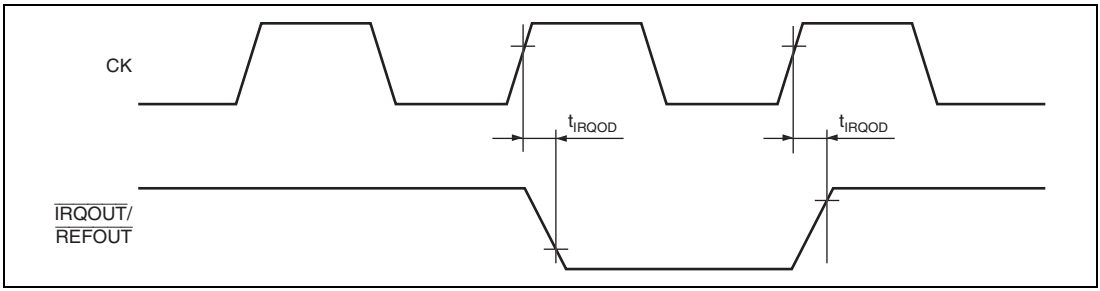


Figure 33.9 Interrupt Signal Output Timing

33.3.3 Bus Timing

Table 33.7 Bus Timing

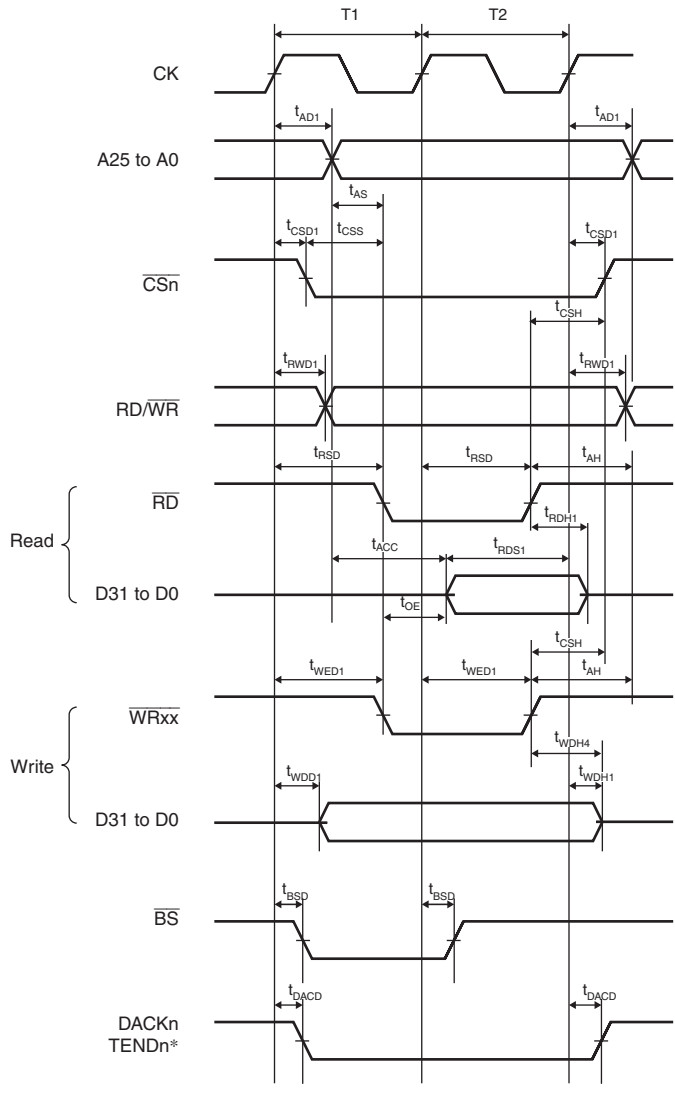
Conditions: $V_{cc}Q = PLLV_{cc} = DrV_{cc} = 3.0$ to 3.6 V, $AV_{cc} = AVREF = 4.5$ V to 5.5 V,
$V_{ss} = PLLV_{ss} = DrV_{ss} = AVREFVSS = AV_{ss} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

		Bϕ = 50 MHz*[1]			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	18	ns	Figures 33.10 to 33.34
Address delay time 2	t_{AD2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figure 33.17
Address delay time 3	t_{AD3}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.35, 33.36
Address setup time	t_{AS}	0	—	ns	Figures 33.10 to 33.13, 33.17
Address hold time	t_{AH}	0	—	ns	Figures 33.10 to 33.13
$\overline{BS}$ delay time	t_{BSD}	—	18	ns	Figures 33.10 to 33.31, 33.35
$\overline{CS}$ delay time 1	t_{CSD1}	1	18	ns	Figures 33.10 to 33.34
$\overline{CS}$ delay time 2	t_{CSD2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.35, 33.36
$\overline{CS}$ setup time	t_{CSS}	0	—	ns	Figures 33.10 to 33.13
$\overline{CS}$ hold time	t_{CSH}	0	—	ns	Figures 33.10 to 33.13
Read write delay time 1	t_{RWD1}	1	18	ns	Figures 33.10 to 33.34
Read write delay time 2	t_{RWD2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.35, 33.36
Read strobe delay time	t_{RSD}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.10 to 33.14, 33.17
Read data setup time 1	t_{RDS1}	$1/2t_{cyc} + 14$	—	ns	Figures 33.10 to 33.14, 33.16
Read data setup time 2	t_{RDS2}	14	—	ns	Figures 33.18 to 33.21, 33.26 to 33.28
Read data setup time 3	t_{RDS3}	$1/2t_{cyc} + 14$	—	ns	Figure 33.17
Read data setup time 4	t_{RDS4}	$1/2t_{cyc} + 14$	—	ns	Figure 33.35

Item	Symbol	Bϕ = 50 MHz*1 Min.	Max.	Unit	Figure
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 33.10 to 33.14, 33.16
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 33.15, 33.18 to 33.21, 33.26 to 33.28
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 33.17
Read data hold time 4	t_{RDH4}	$1/2t_{cyc} + 5$	—	ns	Figure 33.35
Write enable delay time 1	t_{WED1}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.10 to 33.14
Write enable delay time 2	t_{WED2}	—	18	ns	Figure 33.16
Write data delay time 1	t_{WDD1}	—	18	ns	Figures 33.10 to 33.16
Write data delay time 2	t_{WDD2}	—	18	ns	Figures 33.22 to 33.25, 33.29 to 33.31
Write data delay time 3	t_{WDD3}	—	$1/2t_{cyc} + 18$	ns	Figure 33.35
Write data hold time 1	t_{WDH1}	1	15	ns	Figures 33.10 to 33.16
Write data hold time 2	t_{WDH2}	1	—	ns	Figures 33.22 to 33.25, 33.29 to 33.31
Write data hold time 3	t_{WDH3}	$1/2t_{cyc} + 1$	—	ns	Figure 33.35
Write data hold time 4	t_{WDH4}	0	15	ns	Figures 33.10 to 33.14
Read data access time	t_{ACC}*3	t_{cyc} (n + 1.5) – 32*2	—	ns	Figures 33.10 to 33.13
Access time from read strobe	t_{OE}*3	t_{cyc} (n + 1) – 32*2	—	ns	Figures 33.10 to 33.13
$\overline{WAIT}$ setup time	t_{WTS}	$1/2t_{cyc} + 15$	—	ns	Figures 33.11 to 33.17
$\overline{WAIT}$ hold time	t_{WTH}	$1/2t_{cyc} + 2$	—	ns	Figures 33.11 to 33.17
$\overline{RAS}$ delay time 1	t_{RASD1}	1	18	ns	Figures 33.18 to 33.34
$\overline{RAS}$ delay time 2	t_{RASD2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.35, 33.36

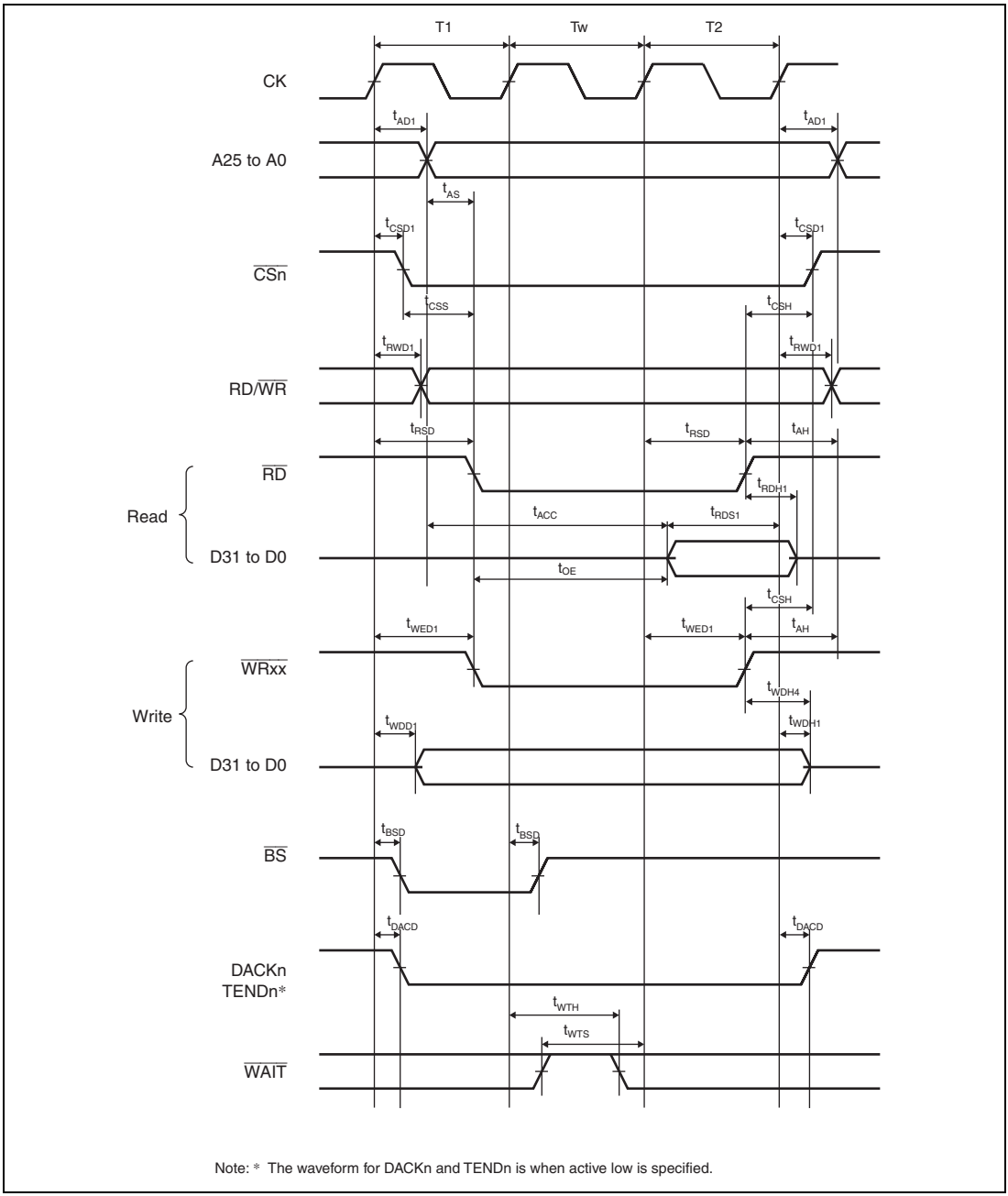
Item	Symbol	Bϕ = 50 MHz*1 Min.	Max.	Unit	Figure
$\overline{CAS}$ delay time 1	t_{CASD1}	1	18	ns	Figures 33.18 to 33.34
$\overline{CAS}$ delay time 2	t_{CASD2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.35, 33.36
DQM delay time 1	t_{DQMD1}	1	18	ns	Figures 33.18 to 33.31
DQM delay time 2	t_{DQMD2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 33.35, 33.36
CKE delay time 1	t_{CKED1}	1	18	ns	Figure 33.33
CKE delay time 2	t_{CKED2}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figure 33.36
$\overline{AH}$ delay time	t_{AHD}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figure 33.14
Multiplexed address delay time	t_{MAD}	–	18	ns	Figure 33.14
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 33.14
DACK, TEND delay time	t_{DACD}	—	Refer to peripheral modules	ns	Figures 33.10 to 33.30, 33.34, 33.38
$\overline{FRAME}$ delay time	t_{FMD}	1	18	ns	Figure 33.15

Notes: 1. The maximum value (f_{max}) of Bϕ (external bus clock) depends on the number of wait cycles and the system configuration of your board.
2. n represents the number of wait cycles.
3. When access-time requirement is satisfied, t_{RDS1} need not be satisfied.



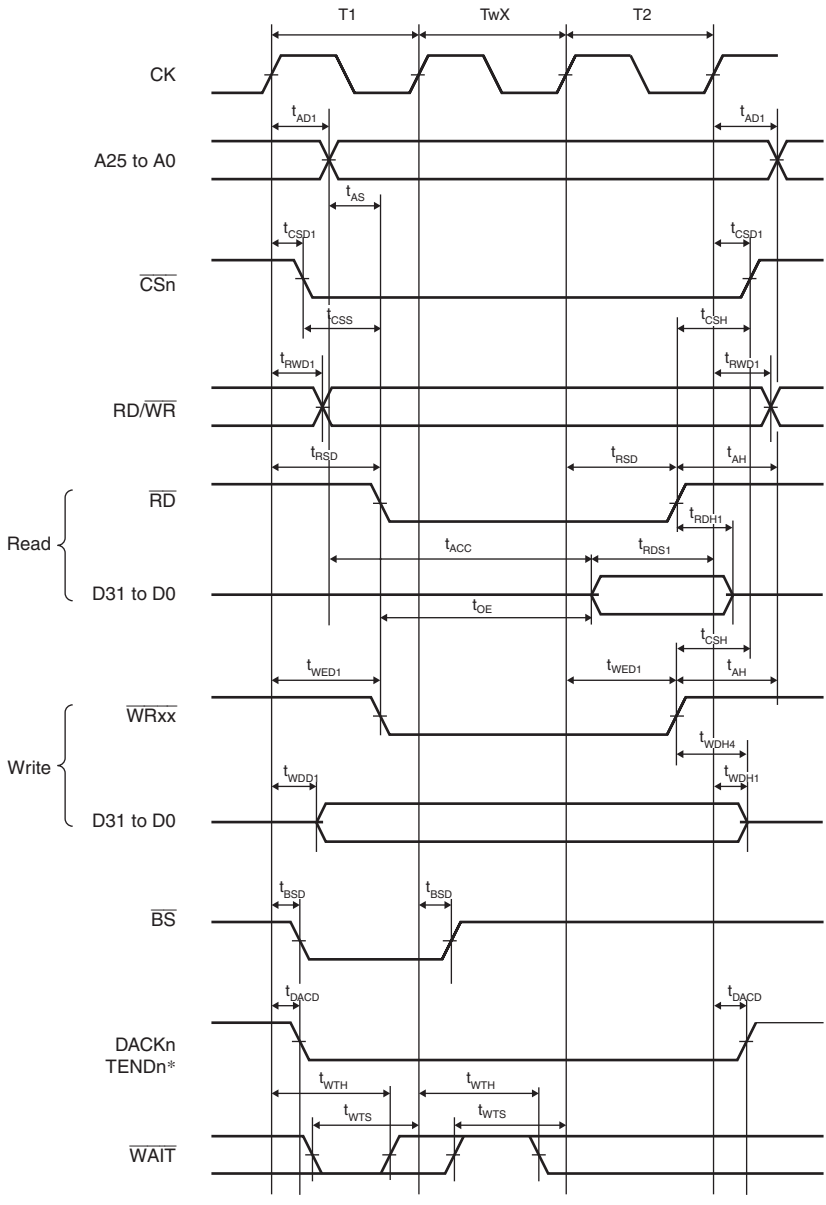
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 33.10 Basic Bus Timing for Normal Space (No Wait)



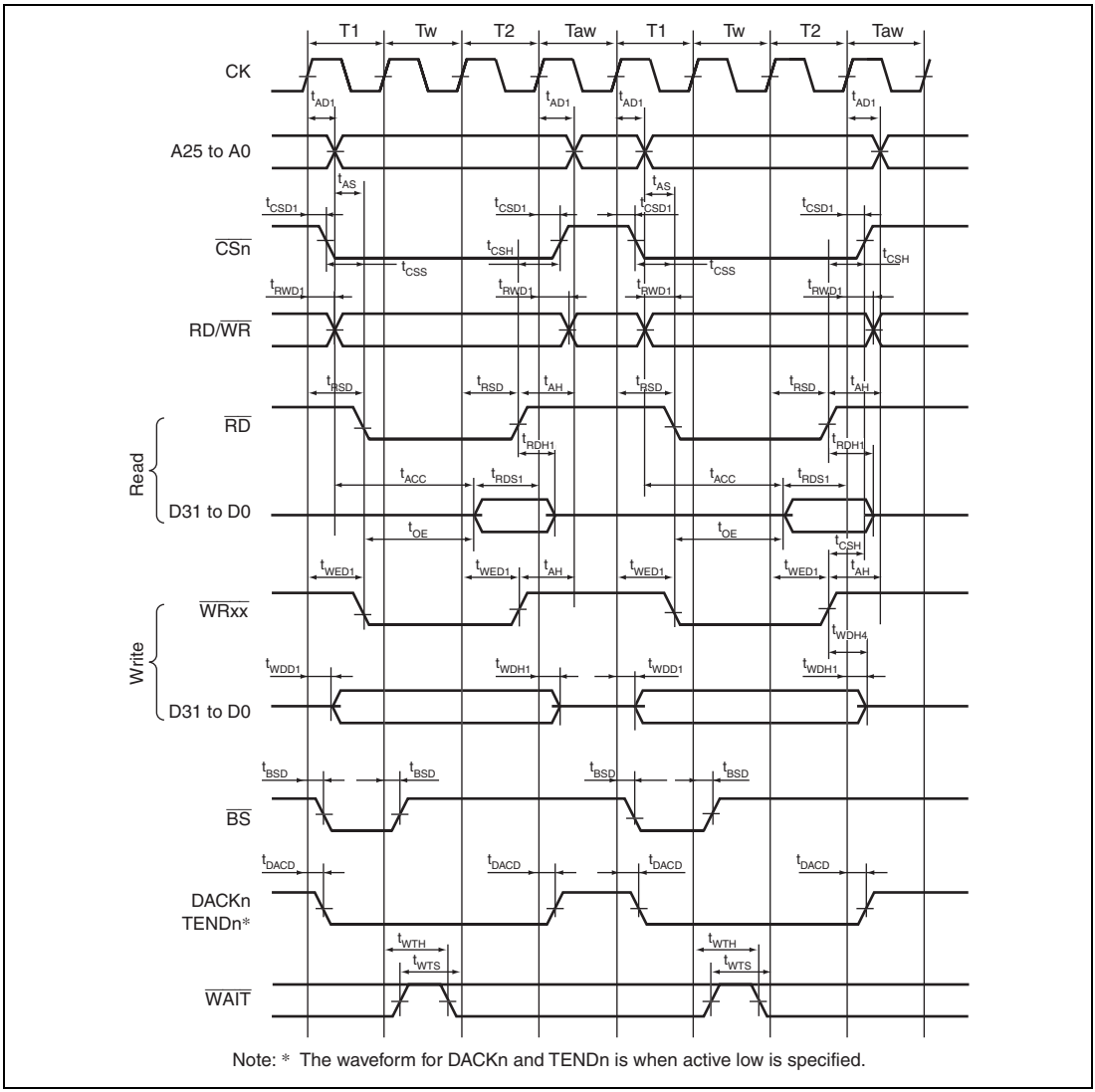
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 33.11 Basic Bus Timing for Normal Space (One Software Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 33.12 Basic Bus Timing for Normal Space (One External Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 33.13 Basic Bus Timing for Normal Space
(One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

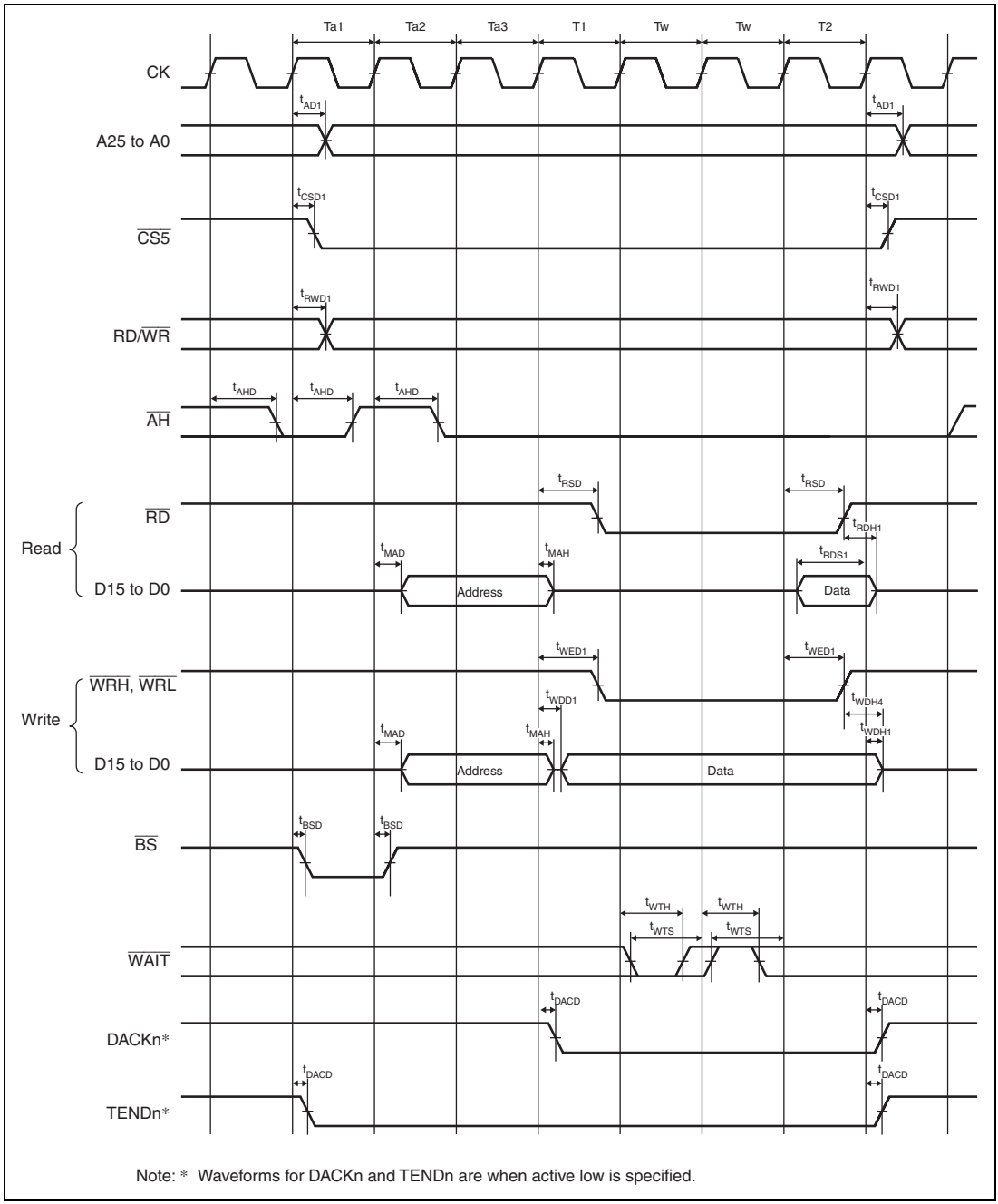
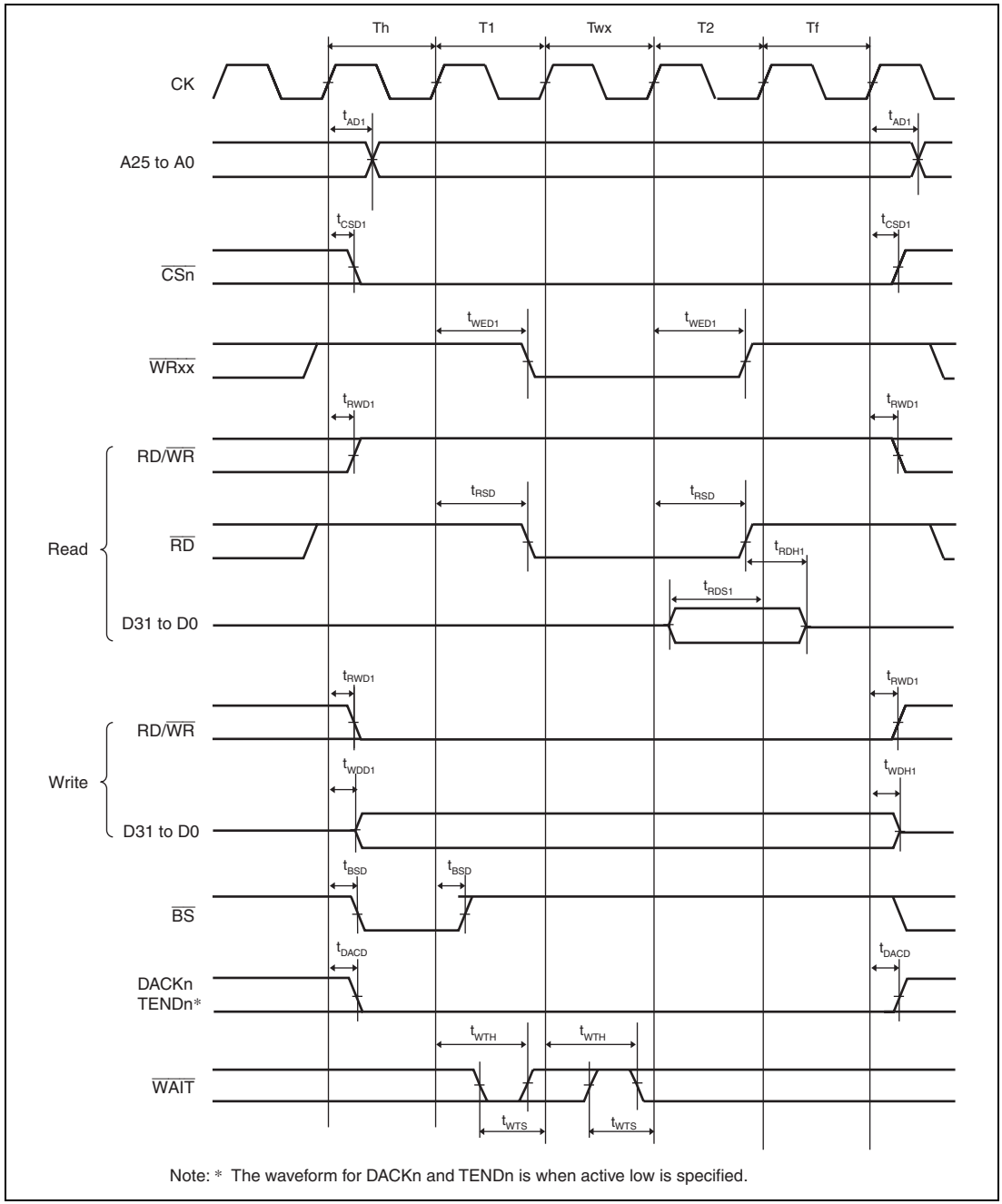


Figure 33.14 MPX-I/O Interface Bus Cycle (Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 33.15 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))

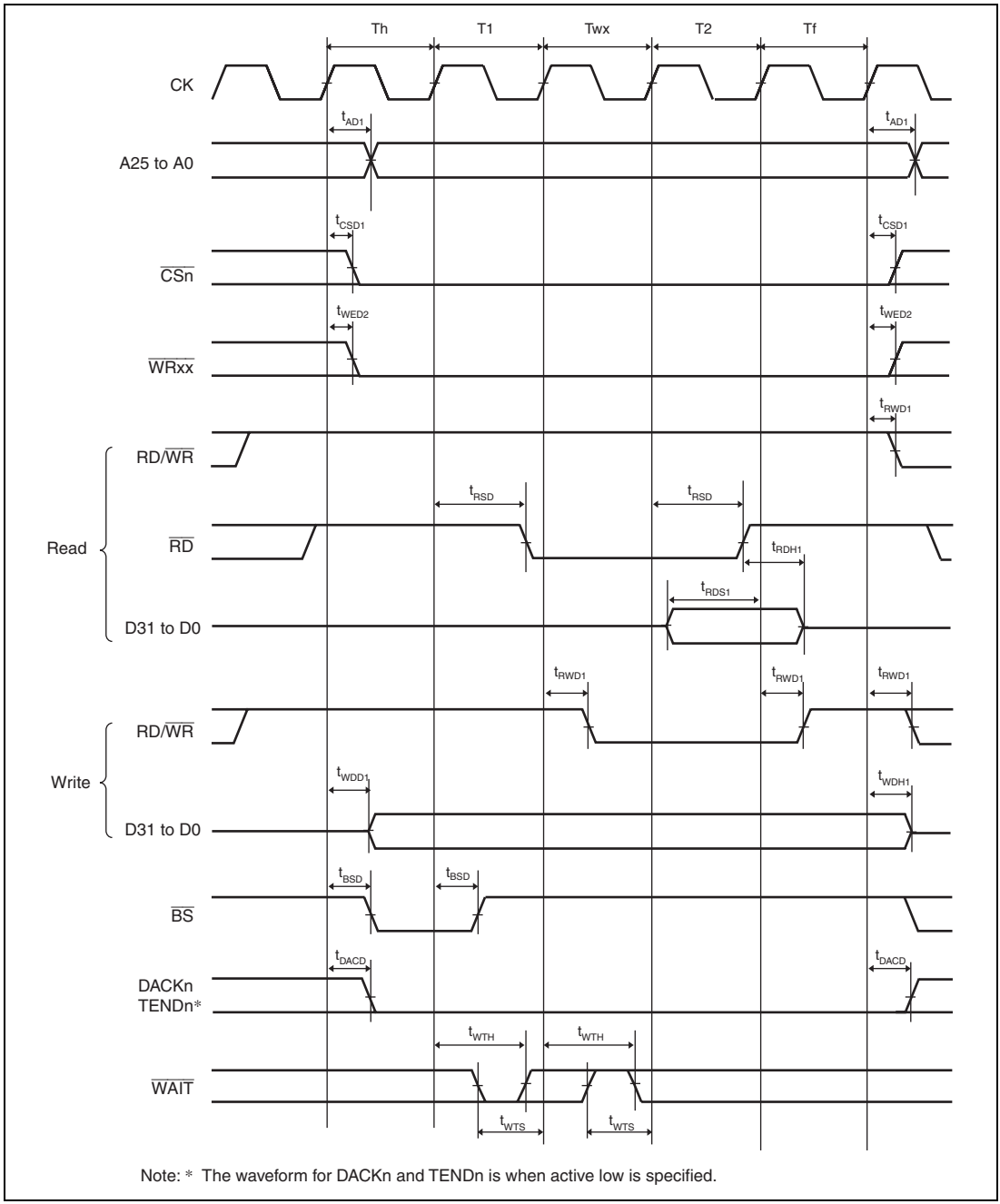
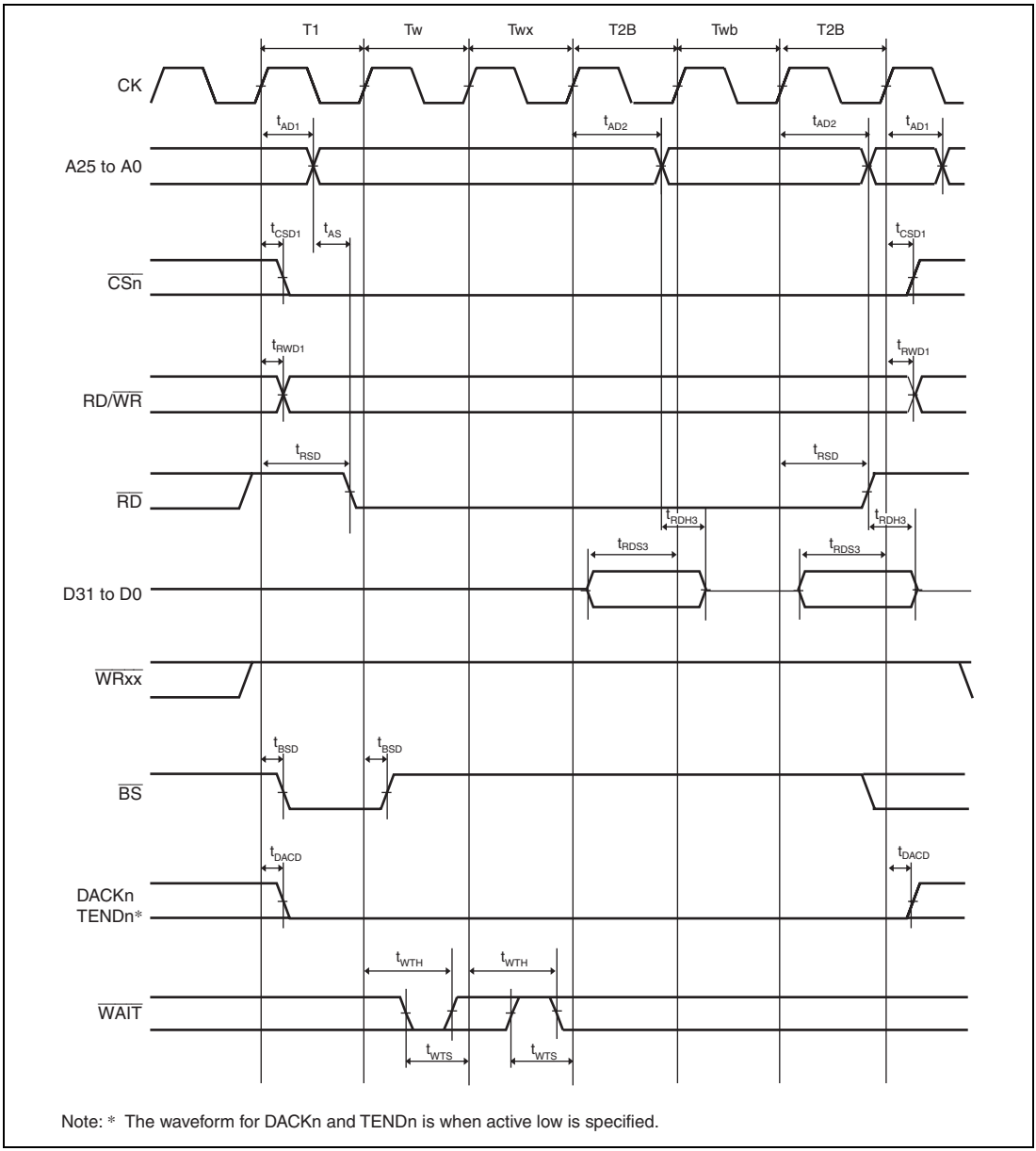
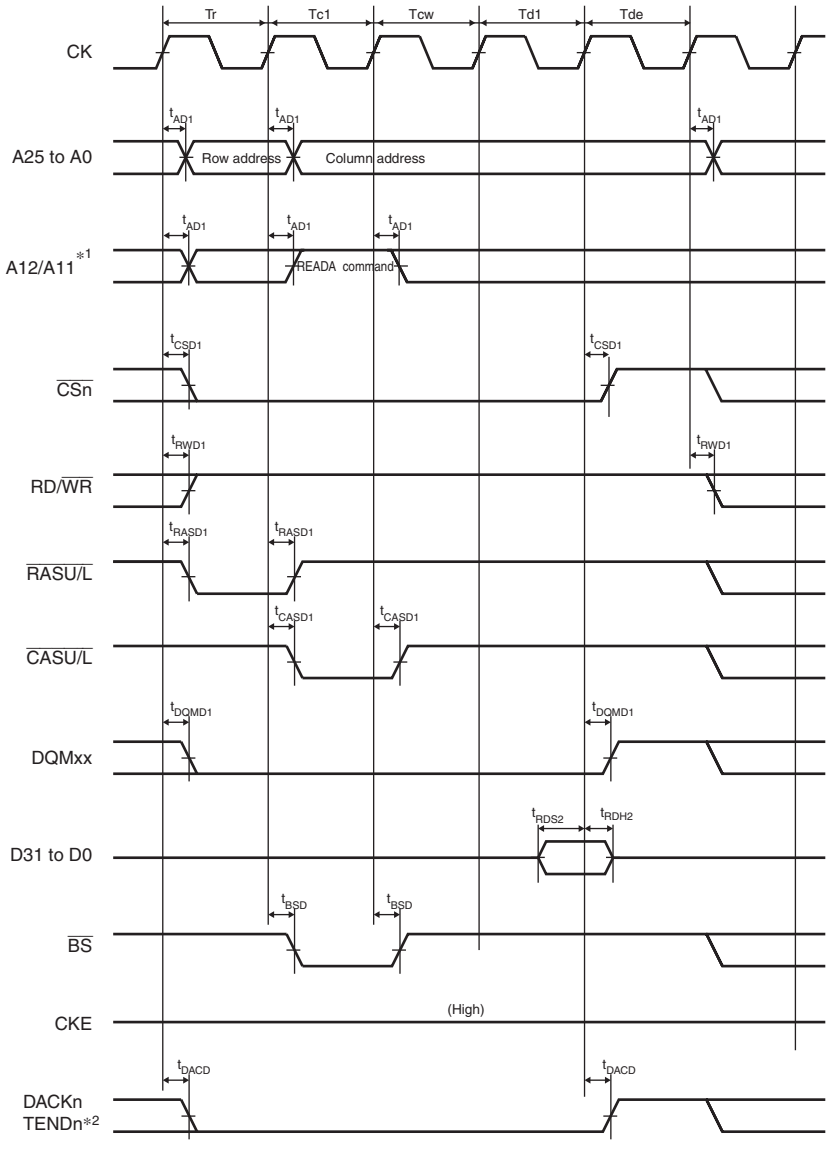


Figure 33.16 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))



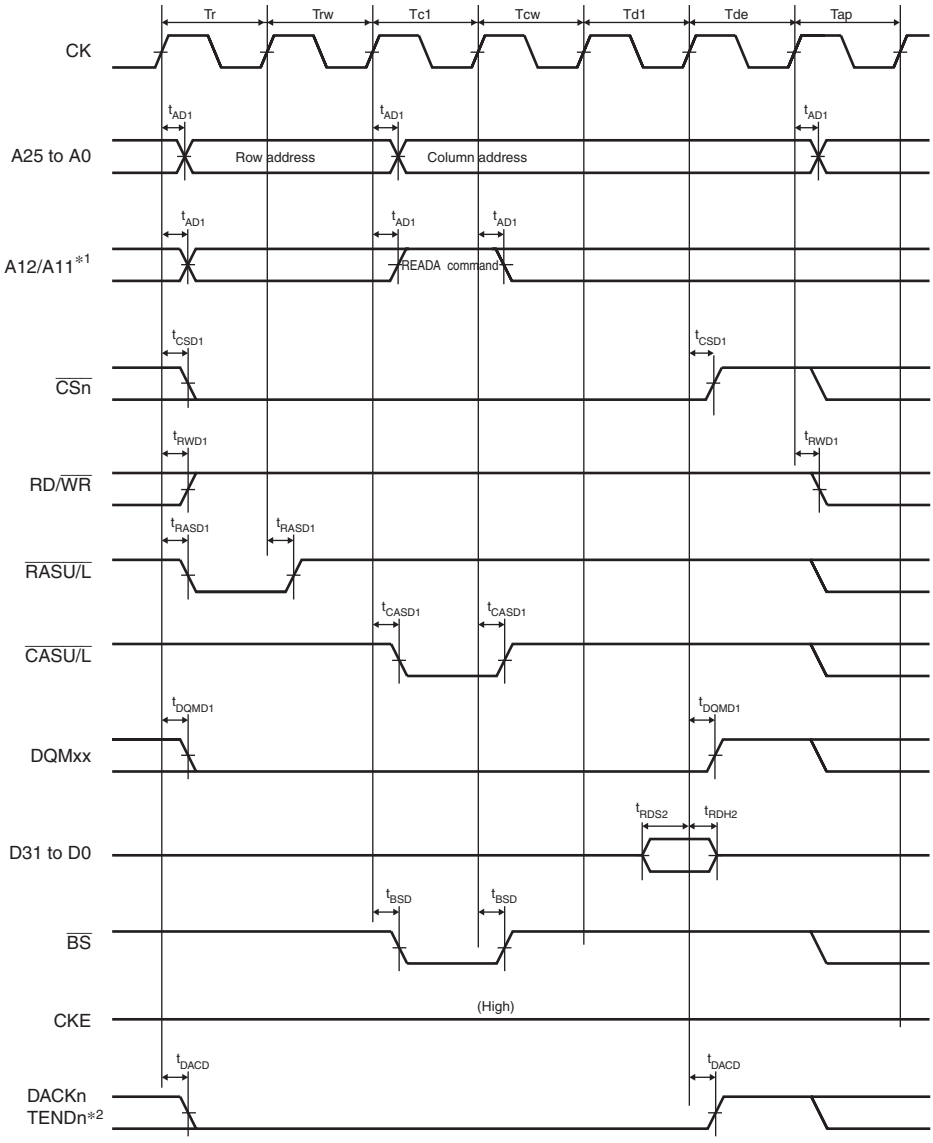
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 33.17 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two-Cycle Burst)



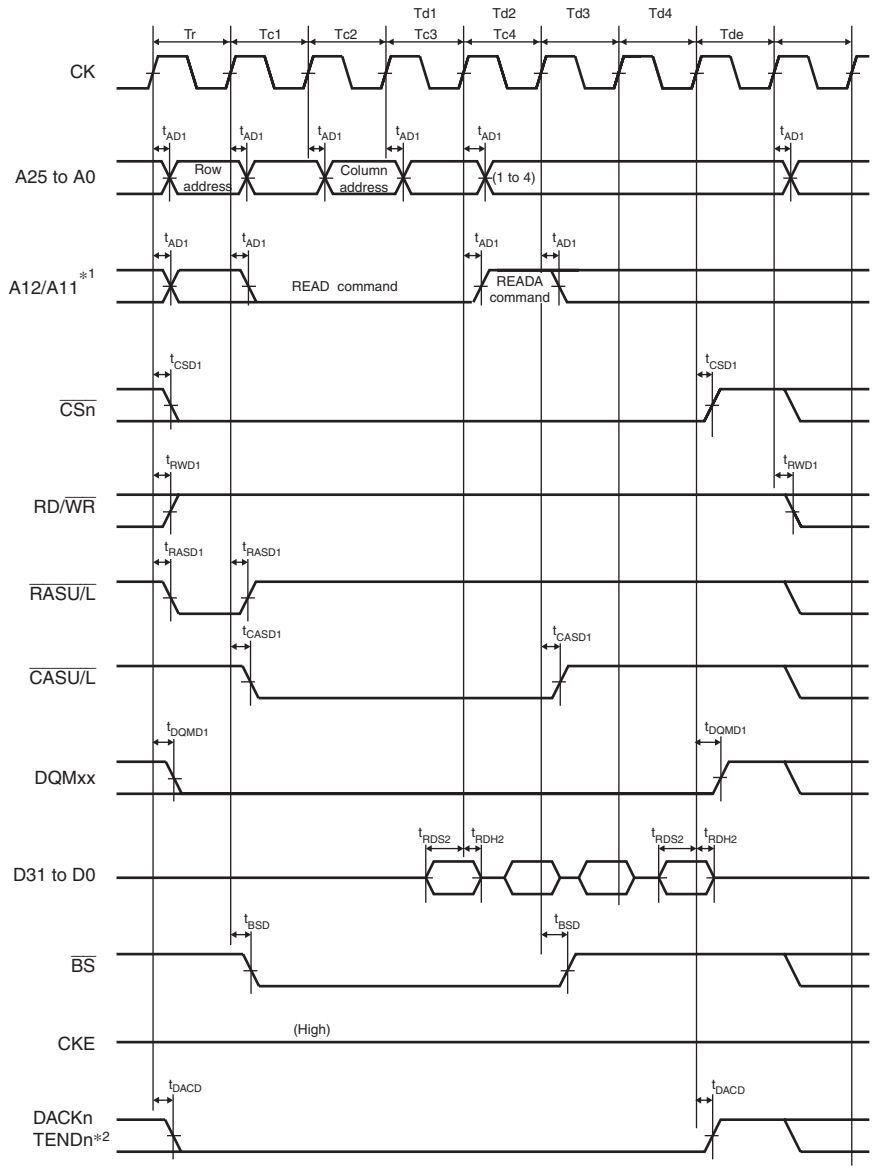
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.18 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



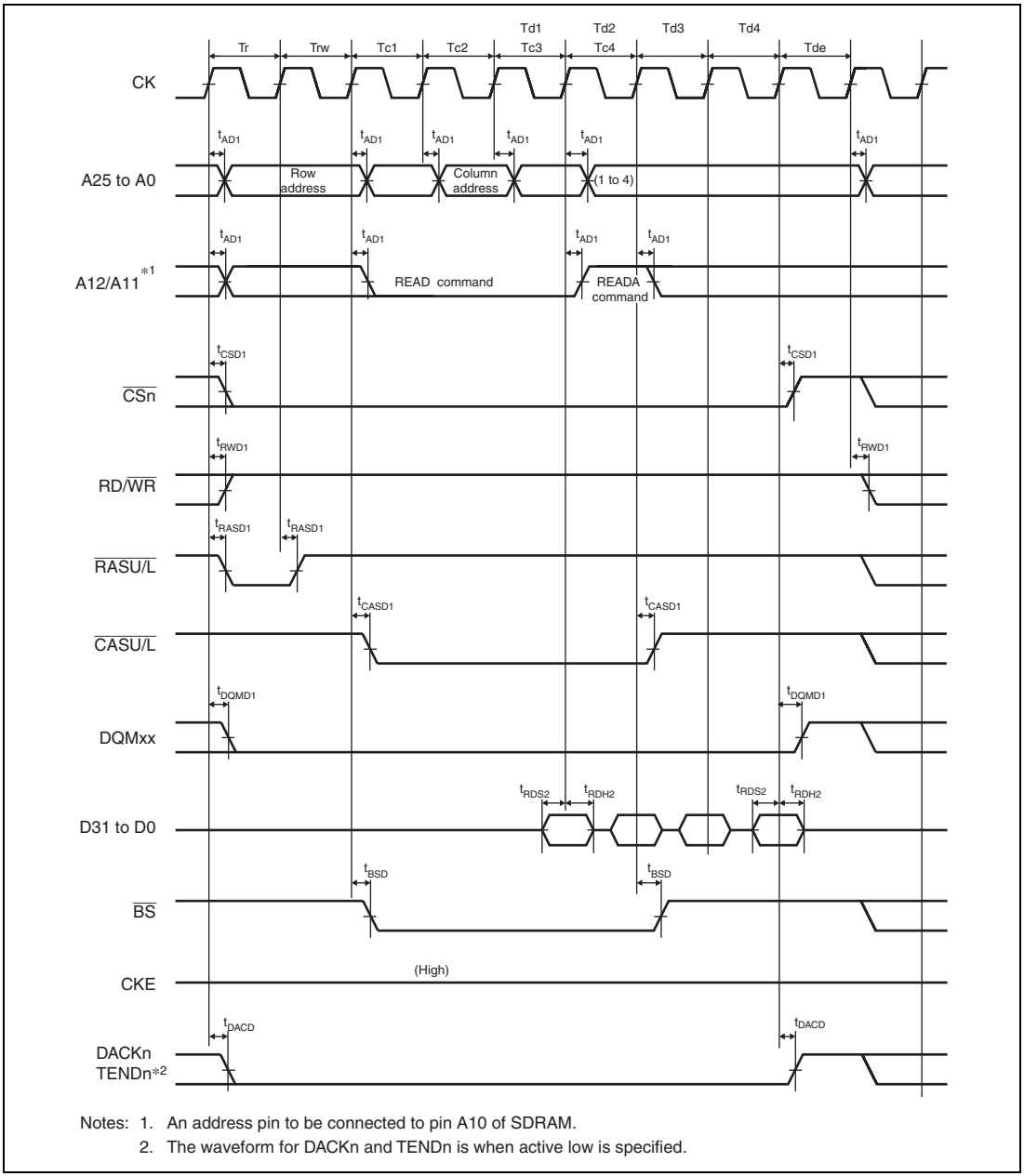
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.19 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



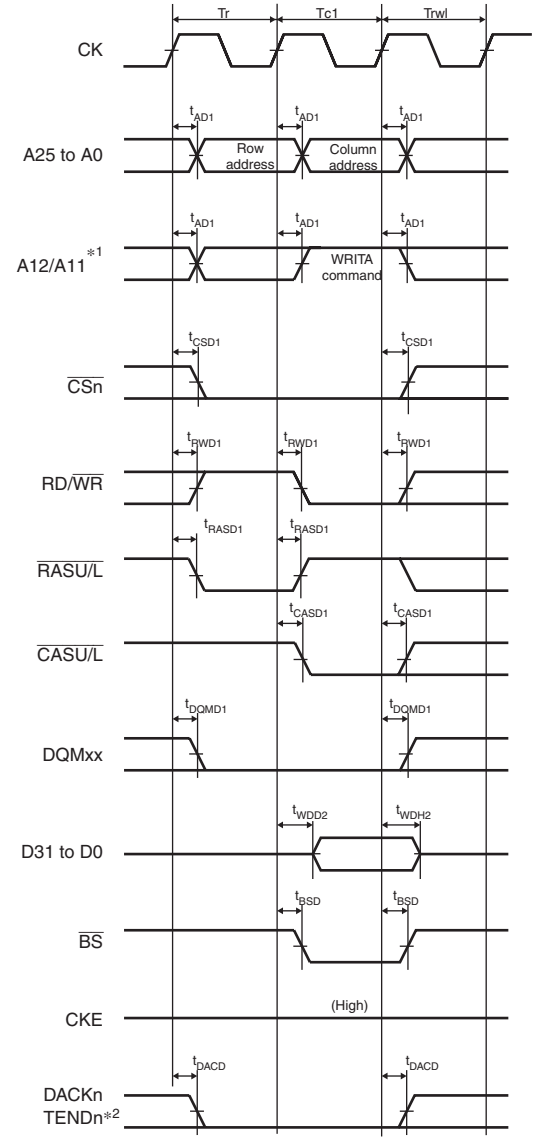
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.20 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



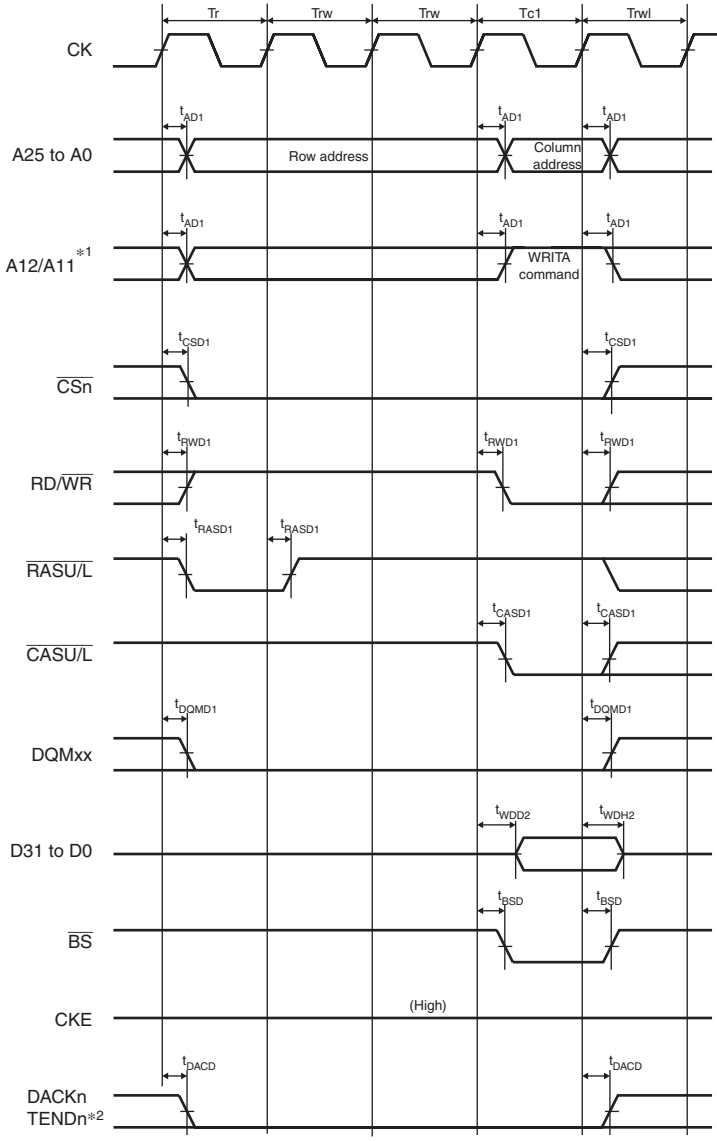
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.21 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



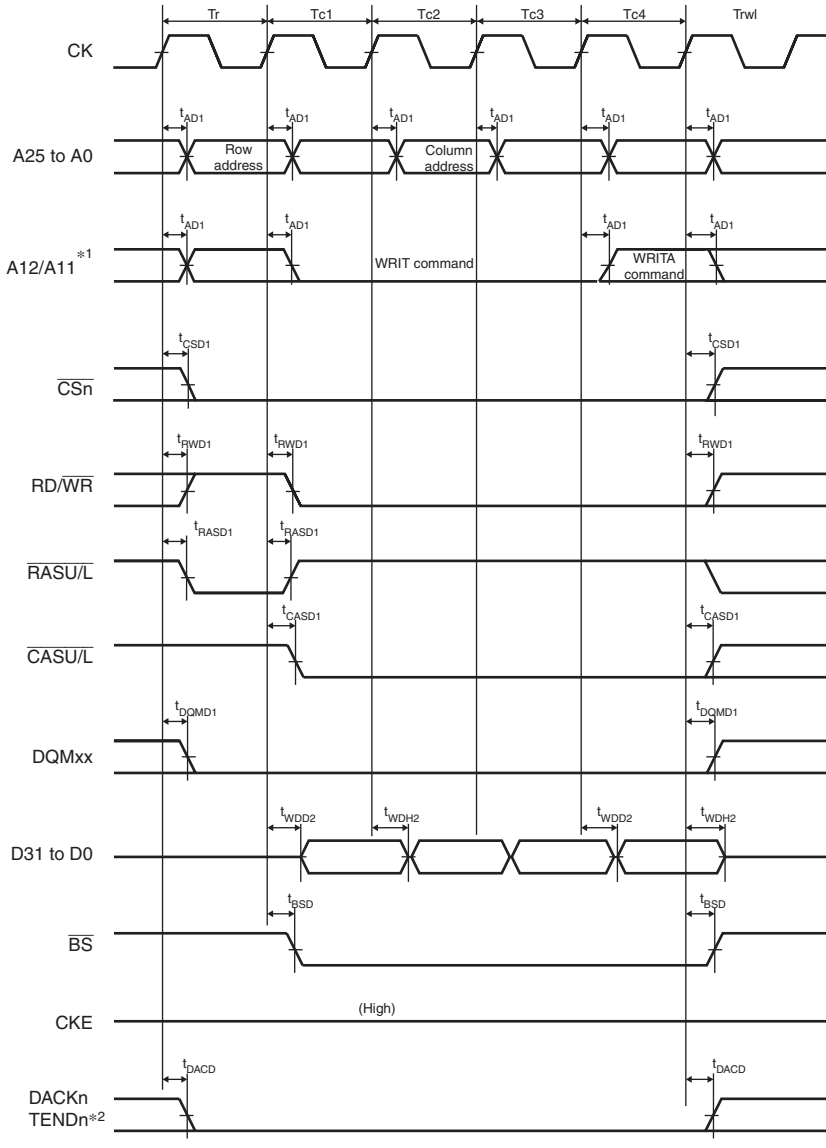
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.22 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)



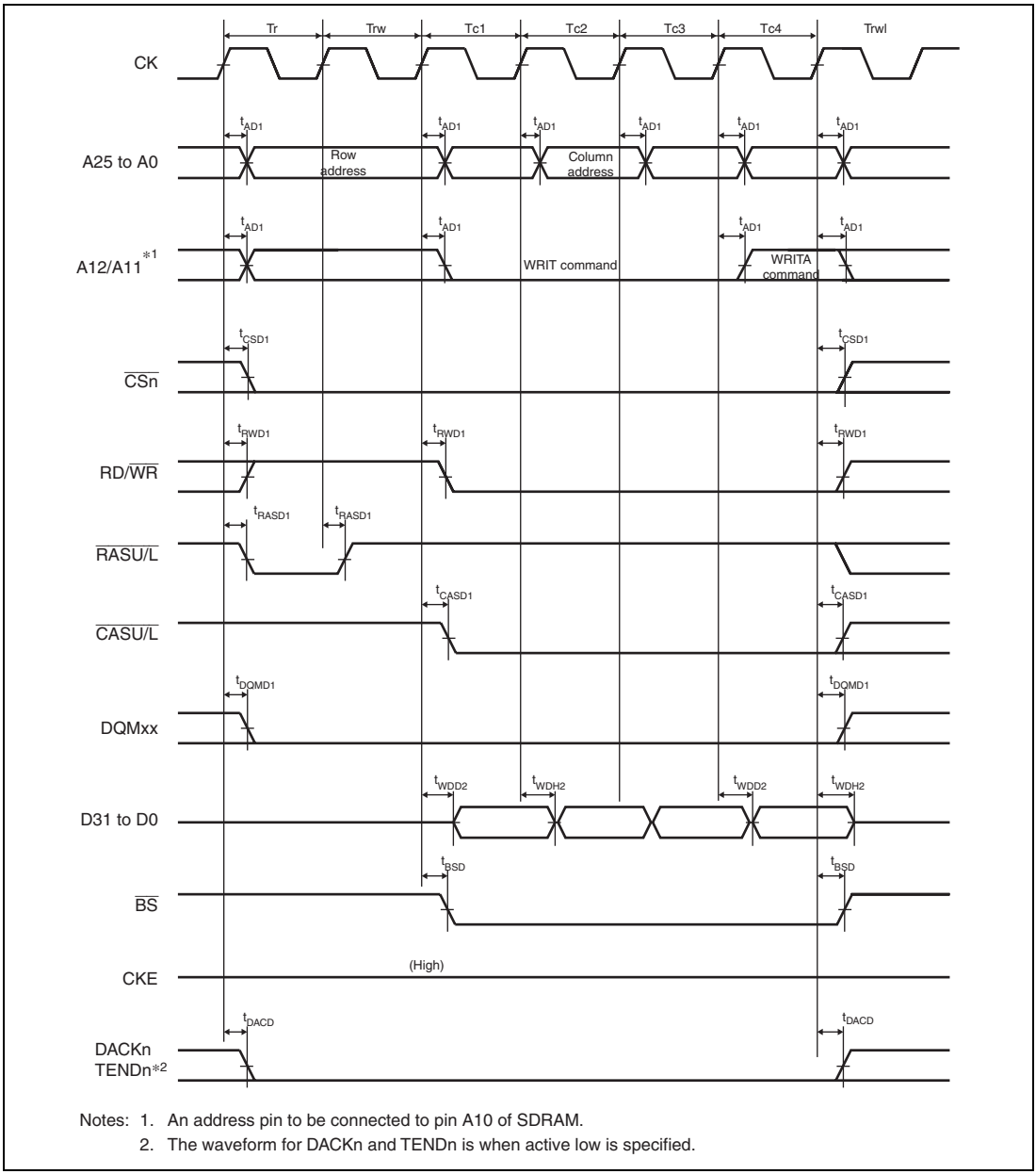
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.23 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



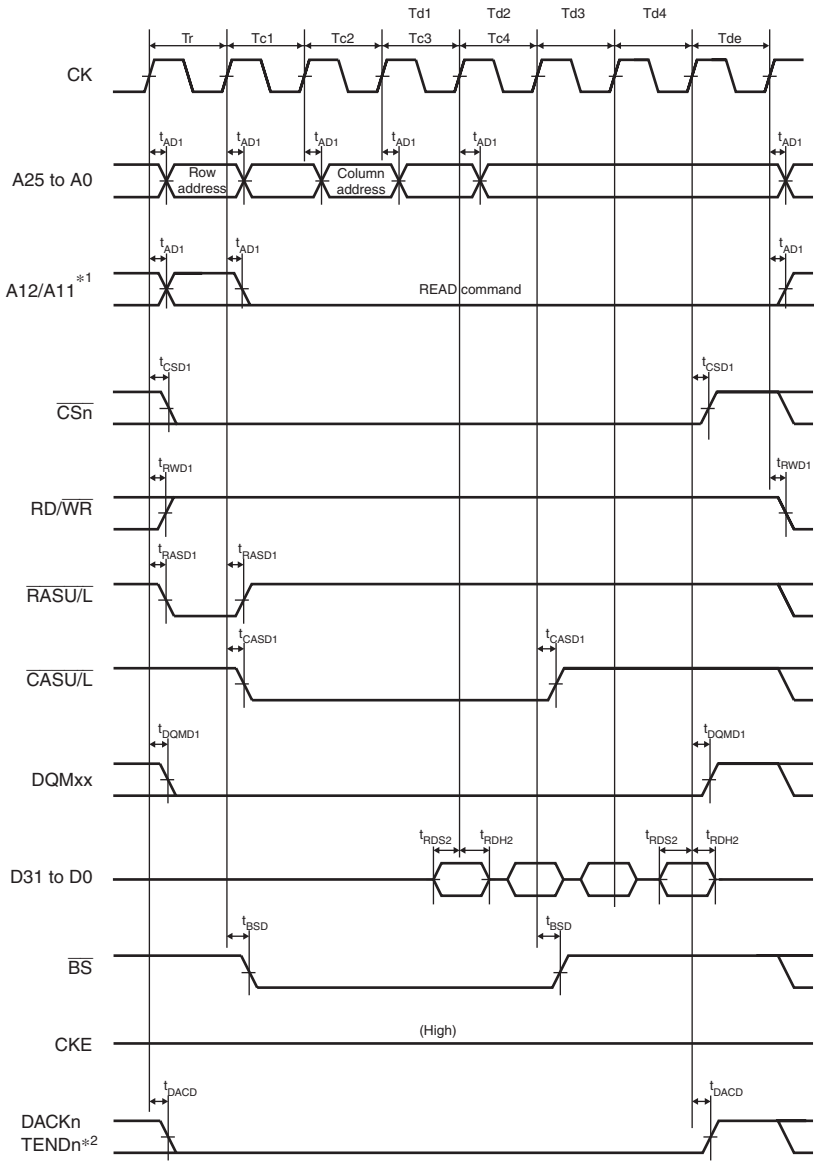
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.24 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



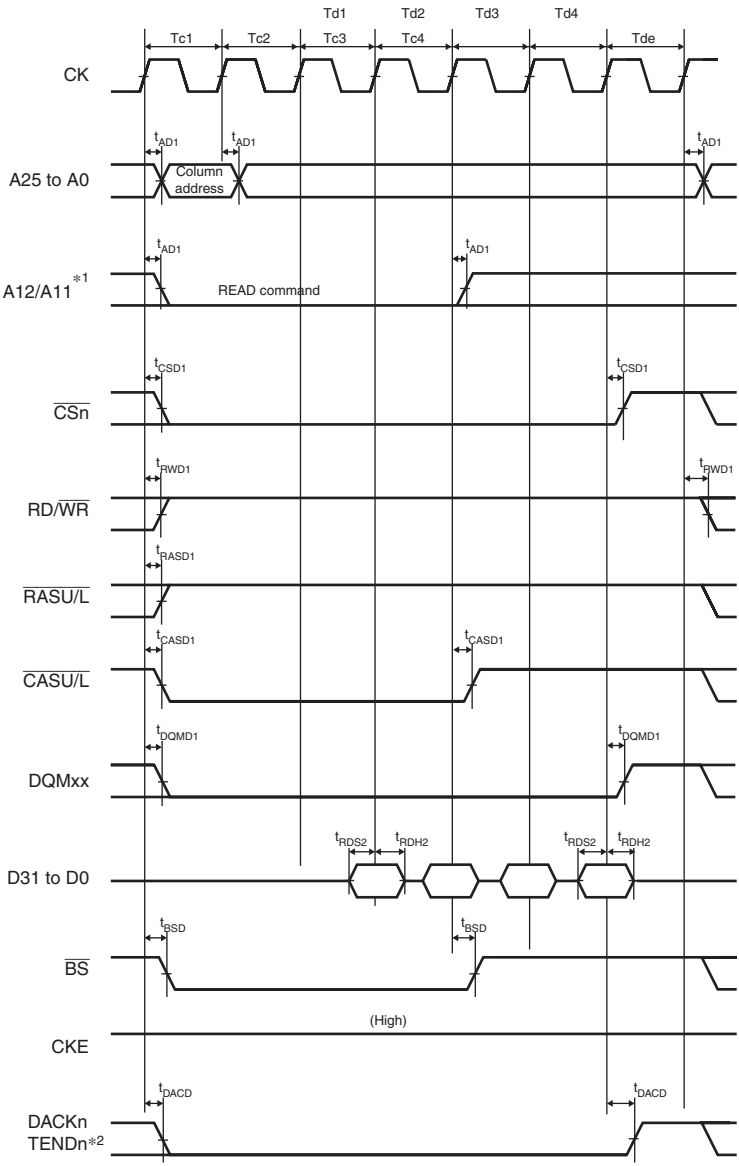
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.25 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



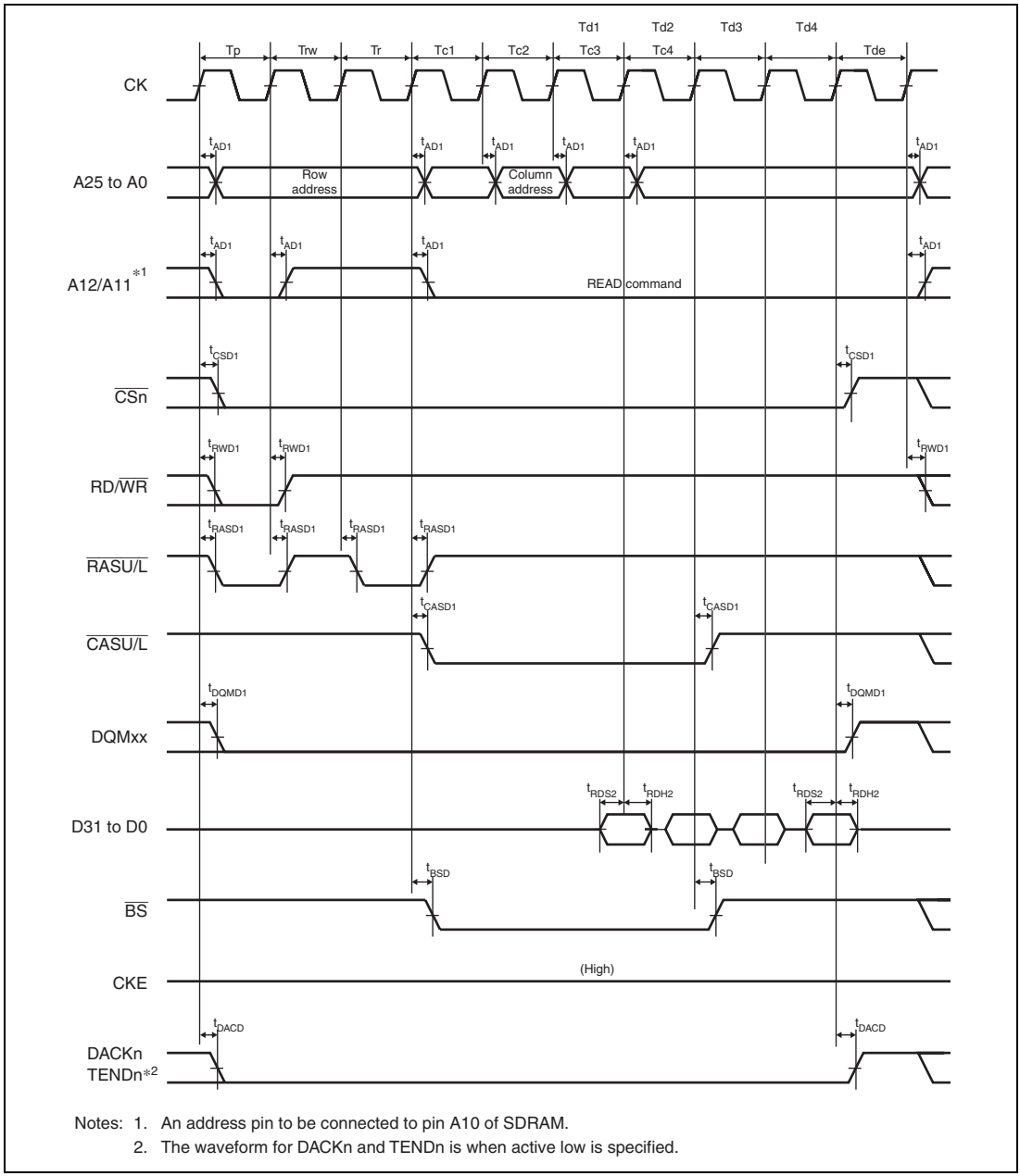
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.26 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)



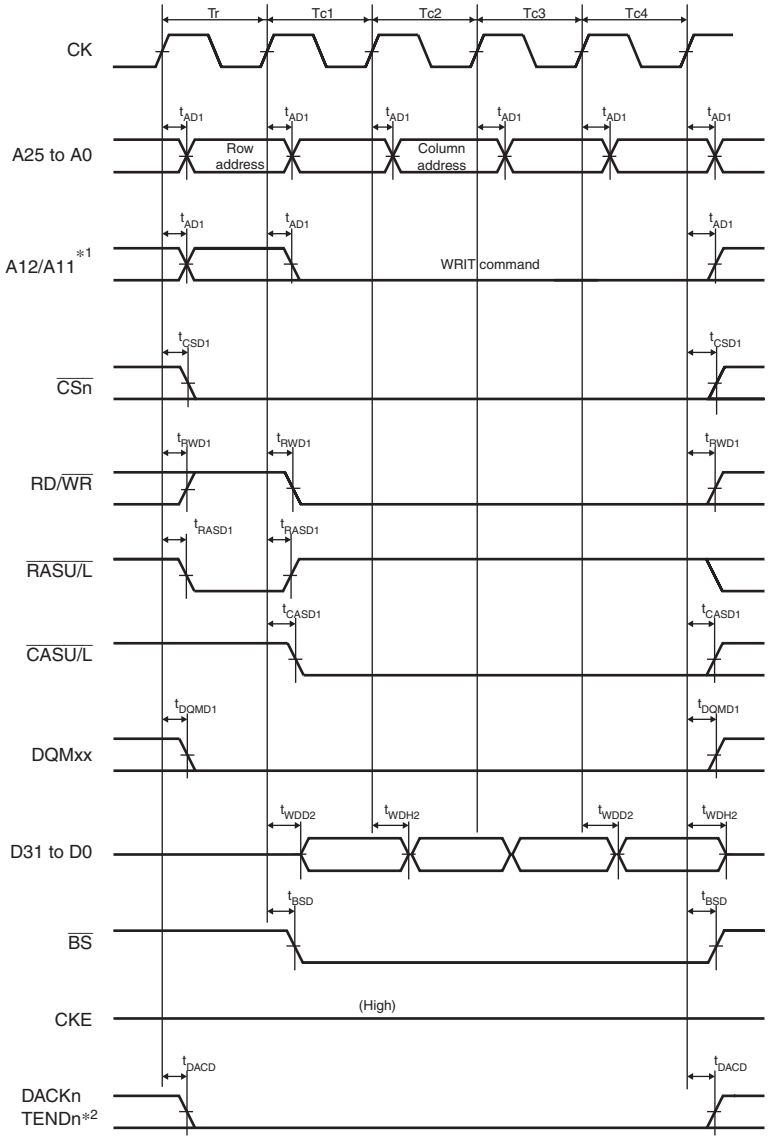
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.27 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)



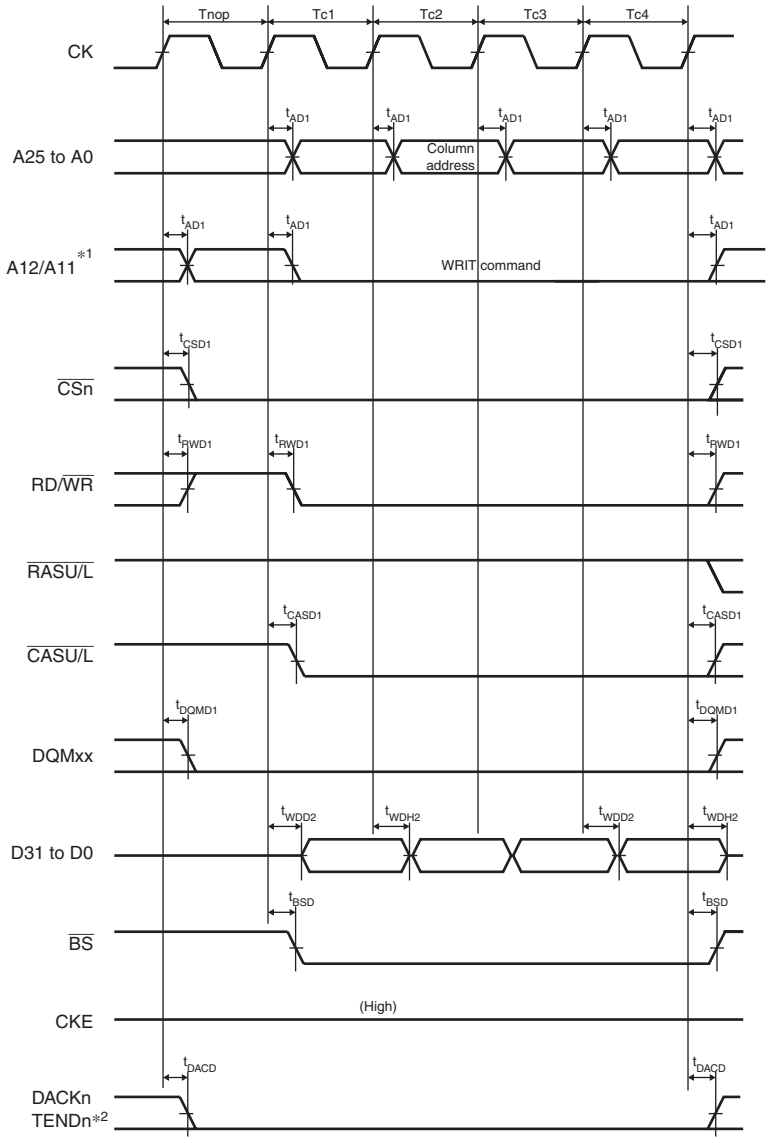
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.28 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency 2, WTRCD = 0 Cycle)



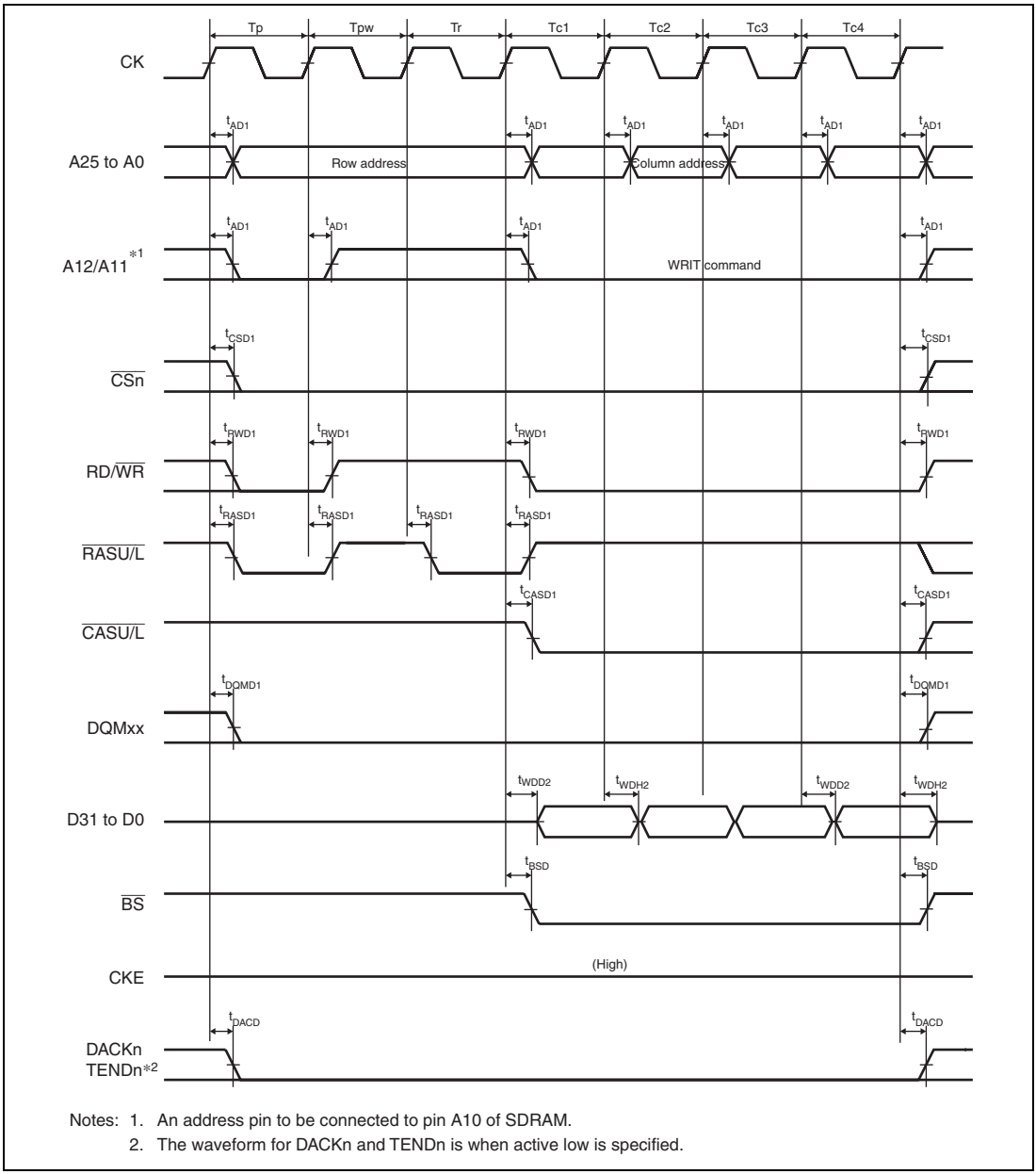
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.29 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



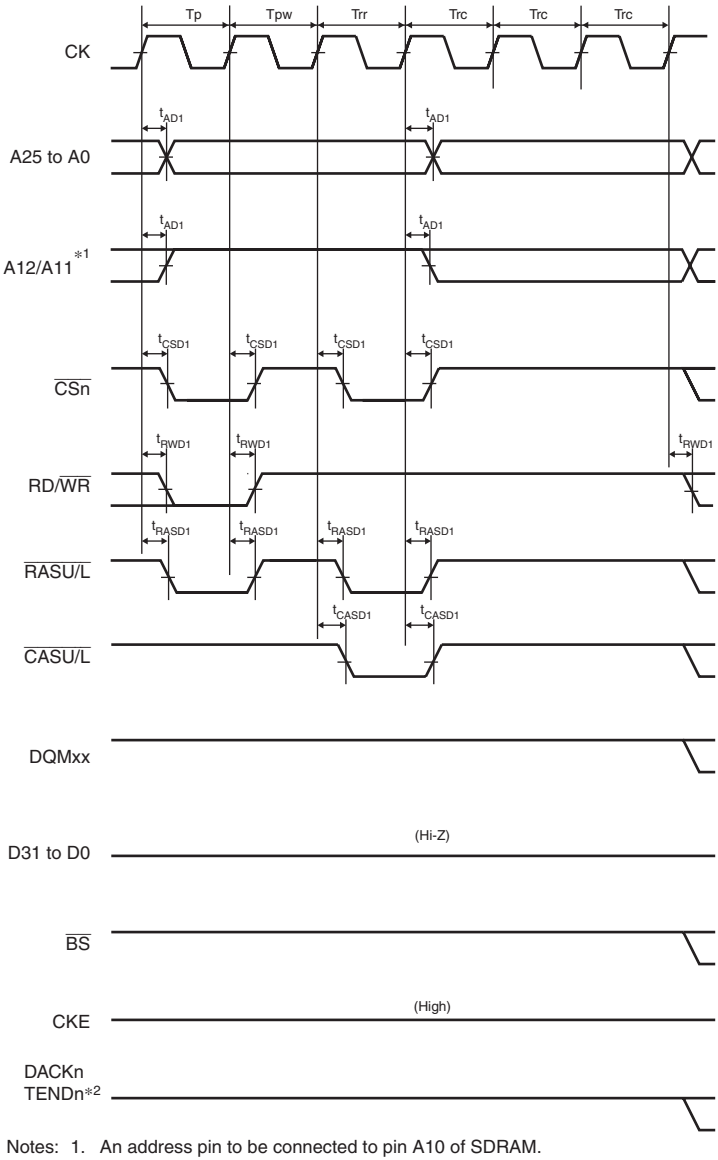
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.30 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)



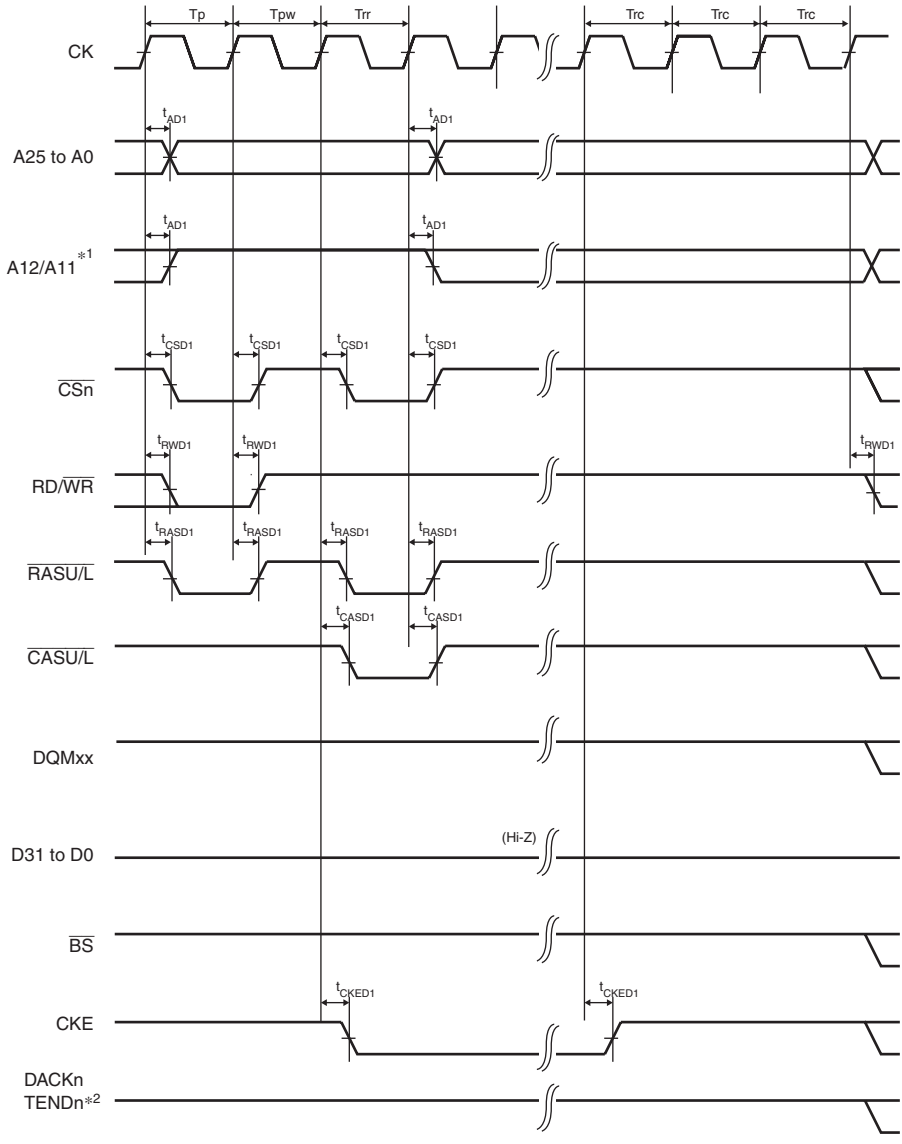
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.31 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)



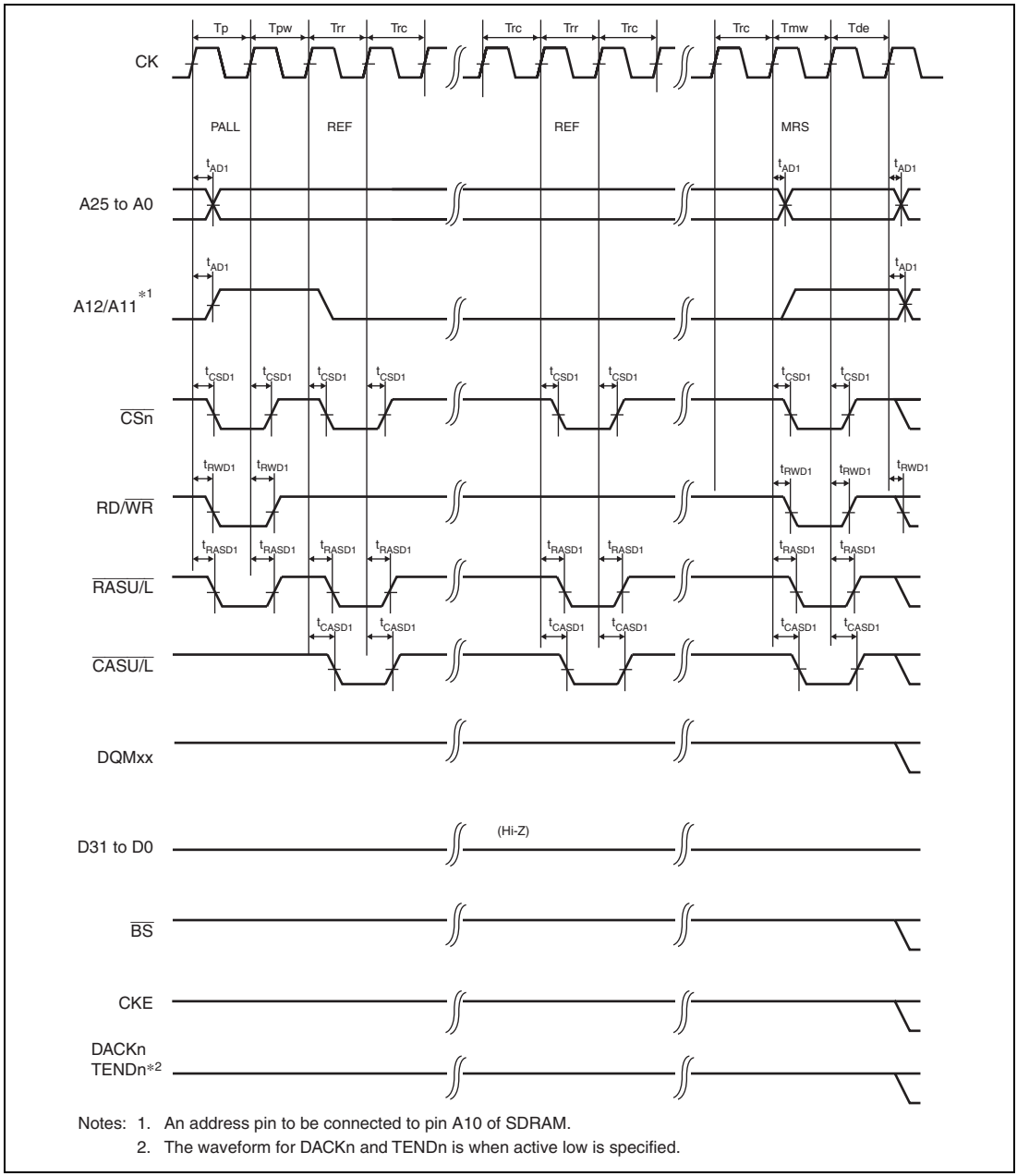
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.32 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



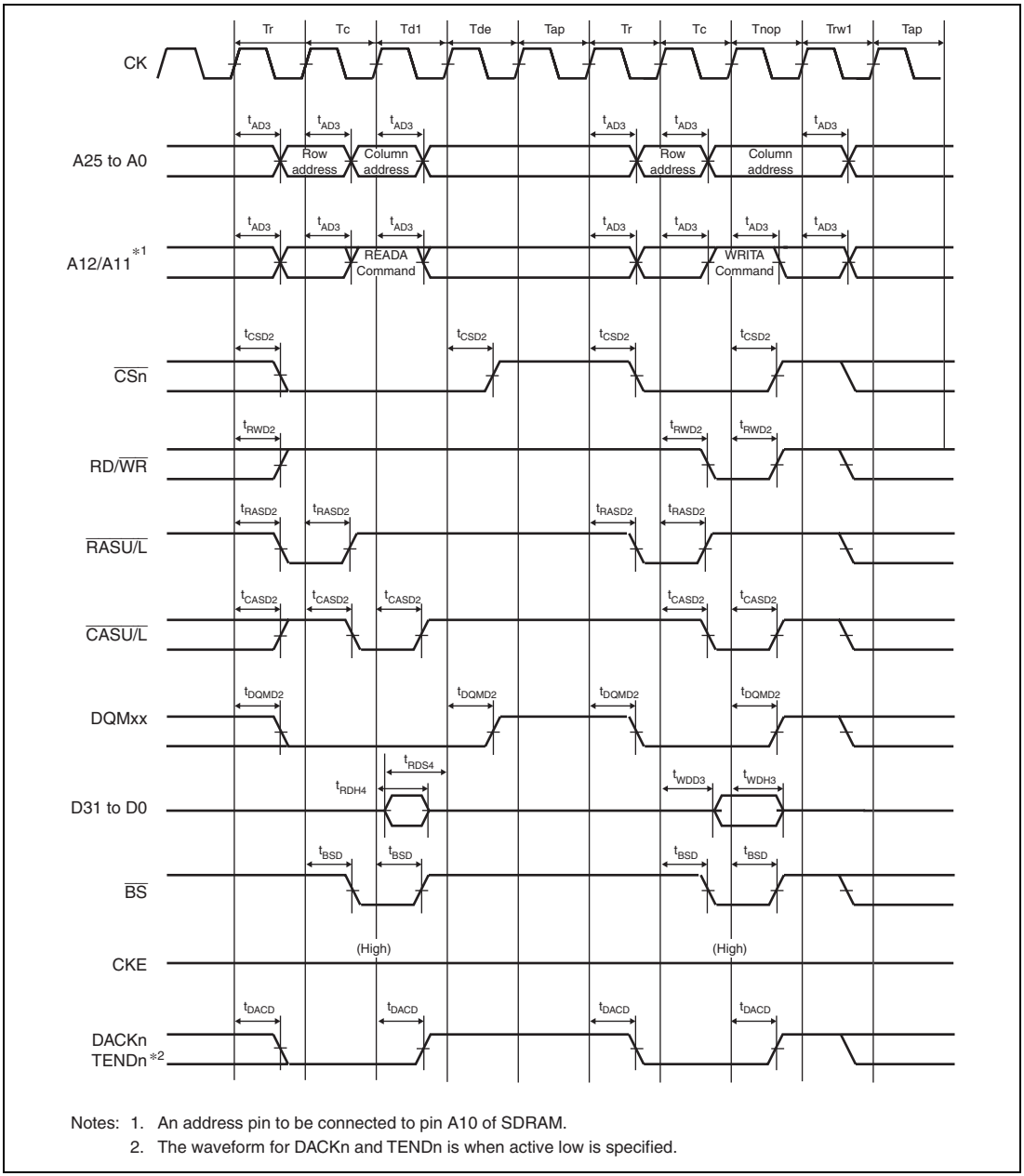
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.33 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



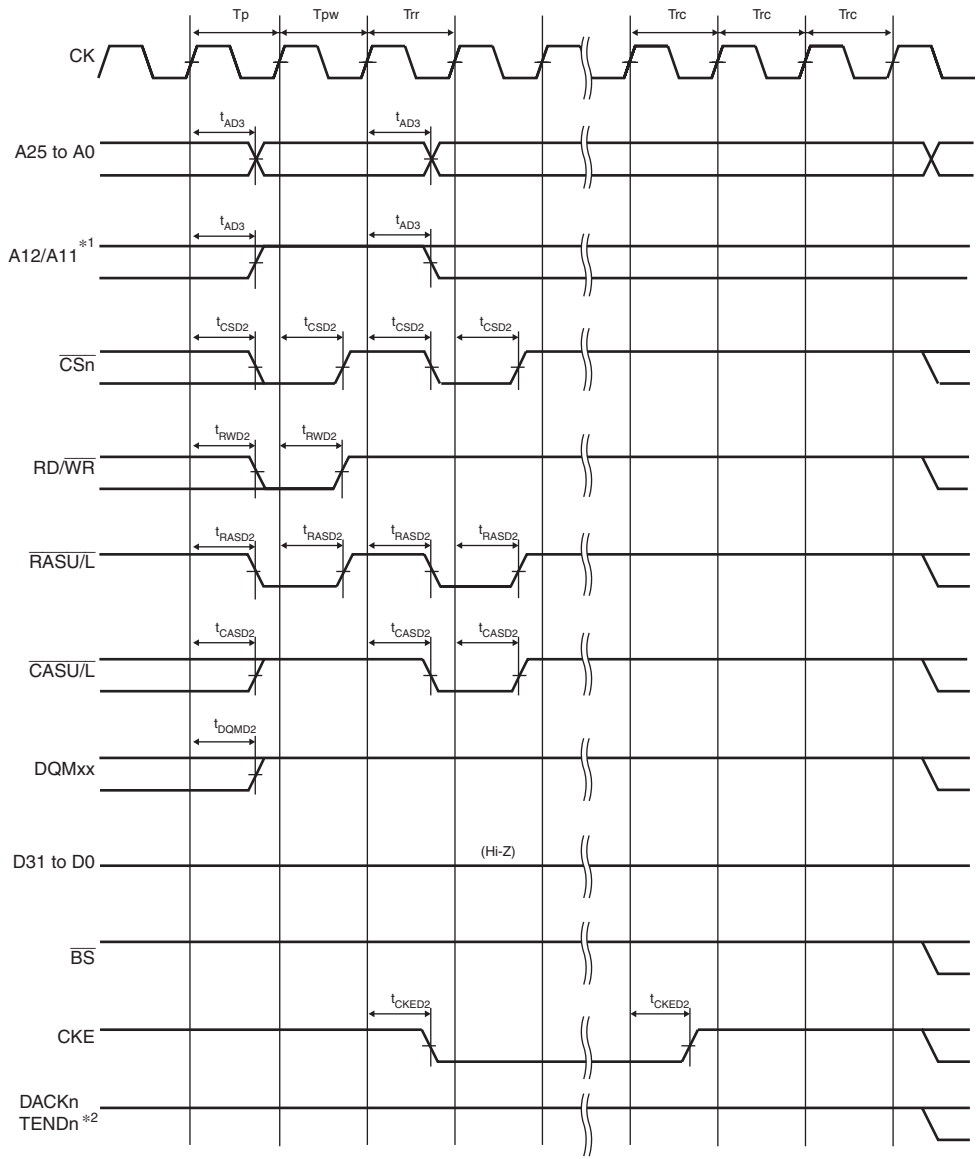
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.34 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.35 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge, TRWL = 2 Cycles)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 33.36 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)

33.3.4 UBC Trigger Timing

Table 33.8 UBC Trigger Timing

Conditions: $V_{cc}Q = PLLV_{cc} = DrV_{cc} = 3.0$ to 3.6 V, $AV_{cc} = AVREF = 4.5$ to 5.5 V,
$V_{ss} = PLLV_{ss} = DrV_{ss} = AVREFVSS = AV_{ss} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	20	ns	Figure 33.37

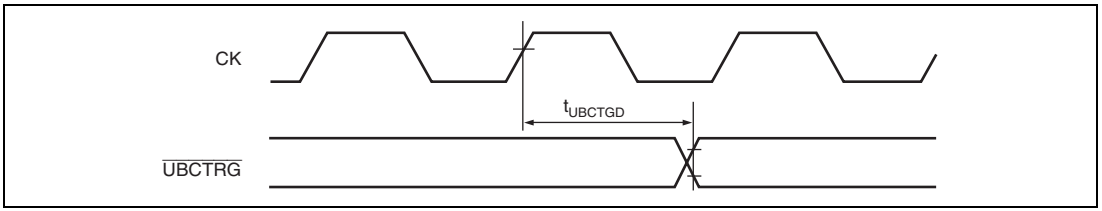


Figure 33.37 UBC Trigger Timing

33.3.5 DMAC Module Timing

Table 33.9 DMAC Module Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC}$= 3.0 to 3.6 V, AV_{CC} = AVREF = 4.5 to 5.5 V,
$V_{SS} = PLLV_{SS} = DrV_{SS}$= AVREFVSS = AV_{SS} = 0 V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	20	—	ns	Figure 33.38
DREQ hold time	t_{DRQH}	20	—		
DACK, TEND delay time	t_{DACD}	—	20		Figure 33.39

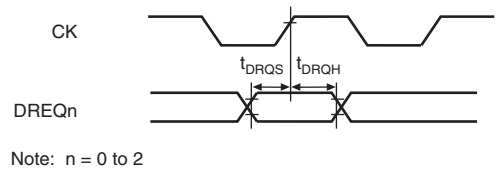


Figure 33.38 DREQ Input Timing

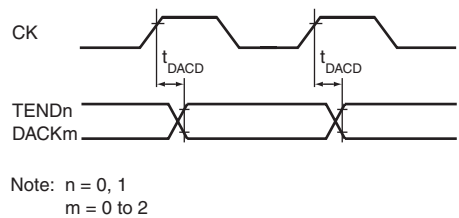


Figure 33.39 DACK, TEND Output Timing

33.3.6 Multi Function Timer Pulse Unit 2 (MTU2) Timing

Table 33.10 Multi Function Timer Pulse Unit 2 (MTU2) Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5 V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	Figure 33.40
Input capture input setup time	t_{TICS}	20	—	ns	
Input capture input pulse width (single edge)	t_{TICW}	1.5	—	t_{Pcyc}	
Input capture input pulse width (both edges)	t_{TICW}	2.5	—	t_{Pcyc}	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 33.41
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{Pcyc}	
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{Pcyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{Pcyc}	

Note: t_{Pcyc} indicates peripheral clock (Pϕ) cycle.

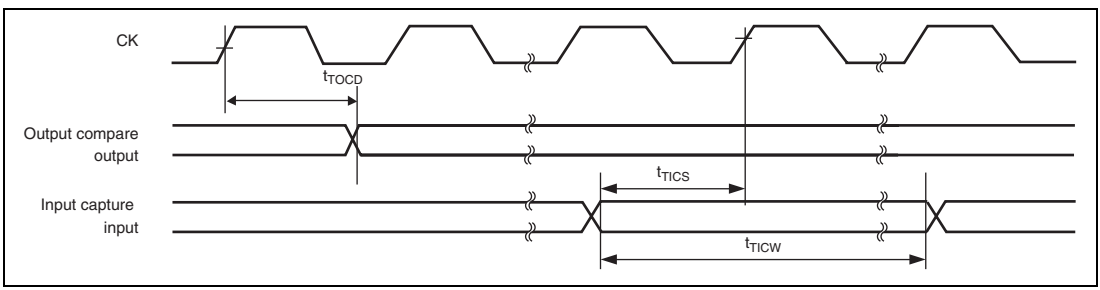


Figure 33.40 MTU2 Input/Output Timing

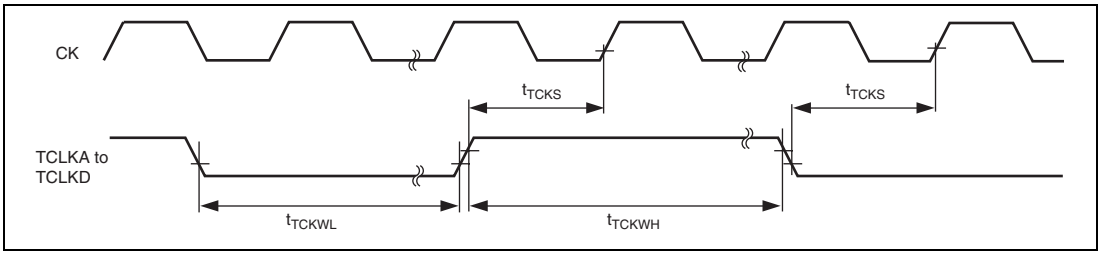


Figure 33.41 MTU2 Clock Input Timing

33.3.7 Multi Function Timer Pulse Unit 2S (MTU2S) Timing

Table 33.11 Multi Function Timer Pulse Unit 2S (MTU2S) Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5 V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	Figure 33.42
Input capture input setup time	t_{TICS}	20	—	ns	
Input capture input pulse width (single edge)	t_{TICW}	1.5	—	t_{Mcyc}	
Input capture input pulse width (both edges)	t_{TICW}	2.5	—	t_{Mcyc}	

Note: t_{Mcyc} indicates MTU2S clock (Mϕ) cycle.

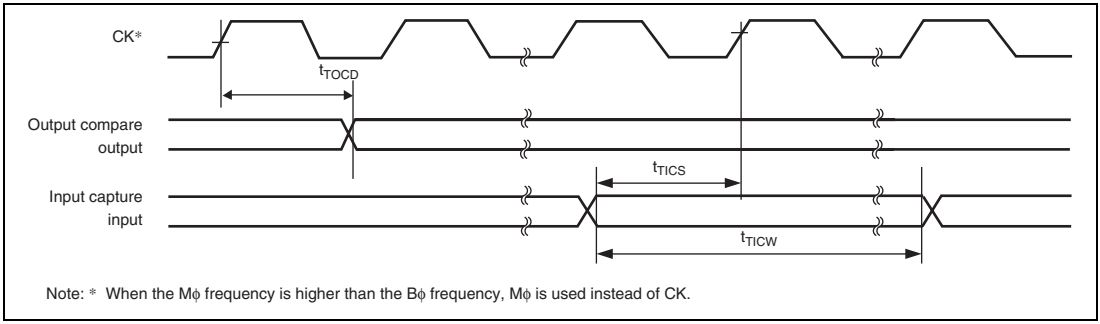


Figure 33.42 MTU2S Input/Output Timing

33.3.8 POE2 Module Timing

Table 33.12 POE2 Module Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5 V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = −40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{POE}}$ input setup time	t_{POES}	50	—	ns	Figure 33.43
$\overline{\text{POE}}$ input pulse width	t_{POEW}	1.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

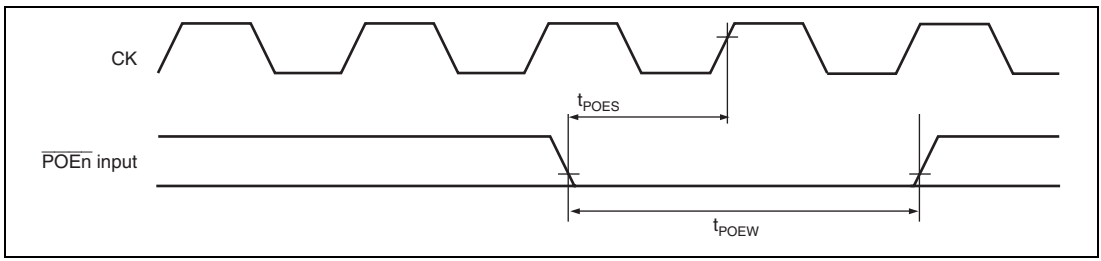


Figure 33.43 POE2 Input/Output Timing

33.3.9 Watchdog Timer Timing

Table 33.13 Watchdog Timer Timing

Conditions: $V_{cc}Q = PLLV_{cc} = DrV_{cc} = 3.0$ to 3.6 V, $AV_{cc} = AVREF = 4.5$ to 5.5 V,
$V_{ss} = PLLV_{ss} = DrV_{ss} = AVREFVSS = AV_{ss} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	50	ns	Figure 33.44

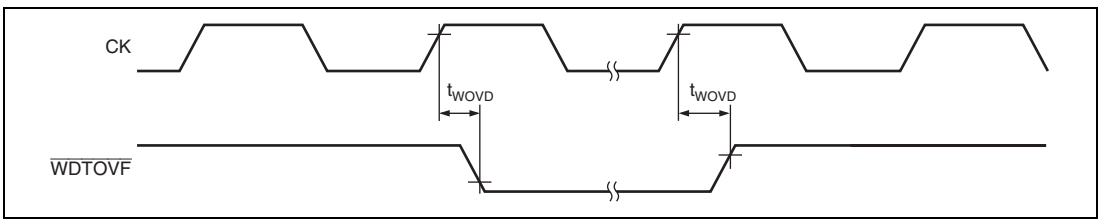


Figure 33.44 Watchdog Timer Timing

33.3.10 Serial Communication Interface (SCI) Timing

Table 33.14 Serial Communication Interface (SCI) Timing

Conditions: $V_{CC}Q$= $PLLV_{CC}$ = DrV_{CC} = 3.0 to 3.6 V, AV_{CC} = AVREF = 4.5 to 5.5 V,
V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AVREFVSS = AV_{SS} = 0 V,
Ta = −40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle (asynchronous)		t_{Scyc}	4	—	t_{pcyc}	Figure 33.45
Input clock cycle (clocked synchronous)		t_{Scyc}	6	—	t_{pcyc}	
Input clock pulse width		t_{SCKW}	0.4	0.6	t_{scyc}	
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	
Transmit data delay time	(asynchronous)	t_{TXD}	—	$4t_{pcyc}$ + 20	ns	Figure 33.46
Receive data setup time		t_{RXS}	$4t_{pcyc}$	—	ns	
Receive data hold time		t_{RXH}	$4t_{pcyc}$	—	ns	
Transmit data delay time	(clocked synchronous)	t_{TXD}	—	$3t_{pcyc}$ + 20	ns	
Receive data setup time		t_{RXS}	$3t_{pcyc}$ + 20	—	ns	
Receive data hold time		t_{RXH}	$3t_{pcyc}$ + 20	—	ns	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

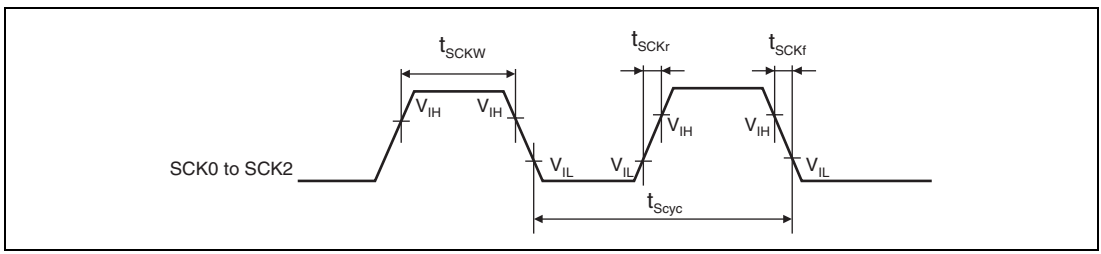


Figure 33.45 Input Clock Timing

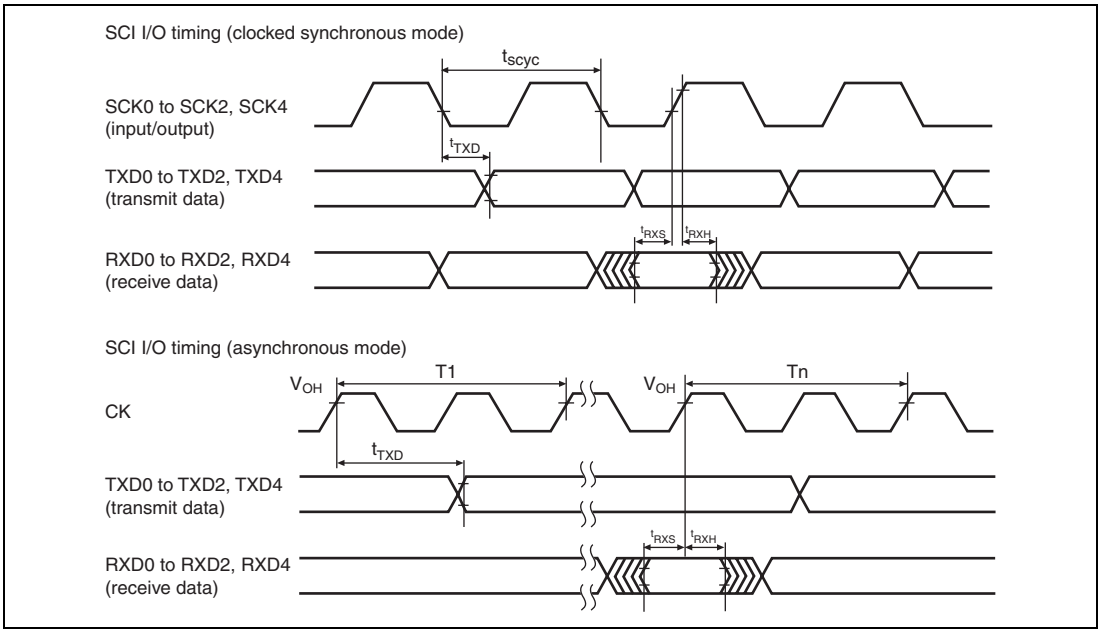


Figure 33.46 SCI Input/Output Timing

33.3.11 SCIF Module Timing

Table 33.15 SCIF Module Timing

Conditions: $V_{CC}Q$= $PLLV_{CC}$ = DrV_{CC} = 3.0 to 3.6 V, AV_{CC} = AVREF = 4.5 to 5.5 V,
V_{SS} = $PLLV_{SS}$ = DrV_{SS} = AVREFVSS = AV_{SS} = 0 V,
Ta = −40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	6	—	t_{pcyc}	Figure 33.47
	(asynchronous)		4	—	t_{pcyc}	
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Transmit data delay time (clocked synchronous)		t_{TXD}	—	$3t_{pcyc}$ + 20	ns	Figure 33.48
Receive data setup time (clocked synchronous)		t_{RXS}	$3t_{pcyc}$ + 20	—	ns	
Receive data hold time (clocked synchronous)		t_{RXH}	$2t_{pcyc}$ + 5	—	ns	
Transmit data delay time (asynchronous)		t_{TXD}	—	$3t_{pcyc}$ + 20	ns	
Receive data setup time (asynchronous)		t_{RXS}	$3t_{pcyc}$ + 20	—	ns	
Receive data hold time (asynchronous)		t_{RXH}	$2t_{pcyc}$ + 5	—	ns	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

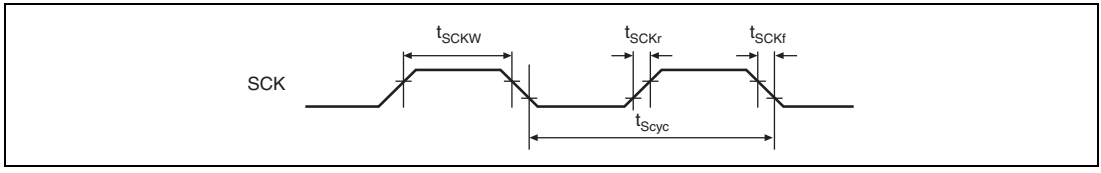


Figure 33.47 SCK Input Clock Timing

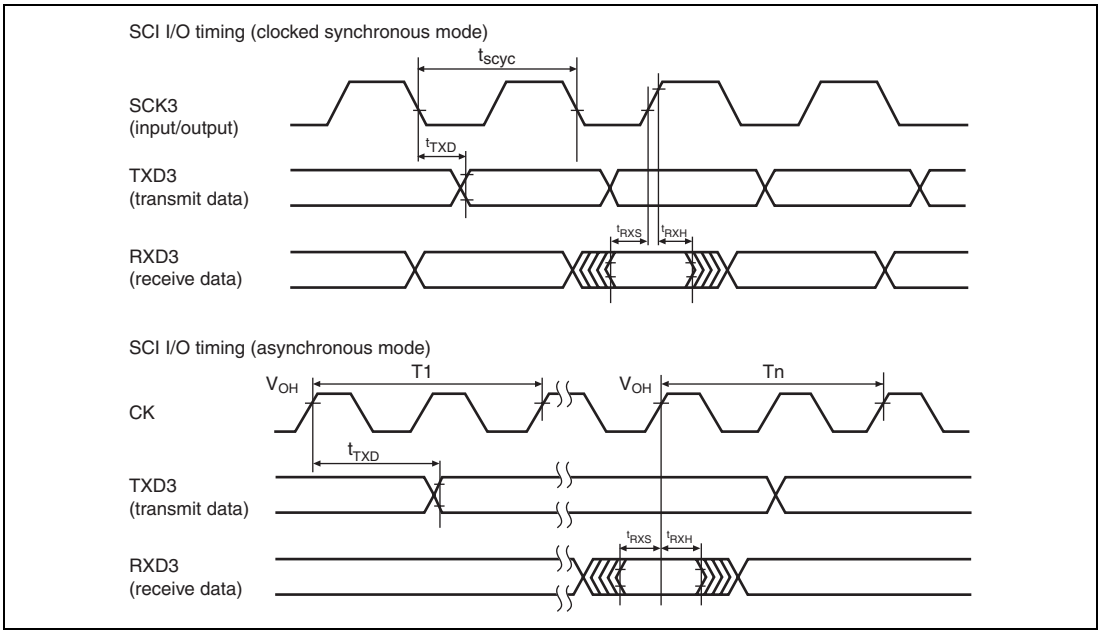


Figure 33.48 SCIF Input/Output Timing

33.3.12 RSPI Timing

Table 33.16 SPI Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFV_{SS} = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Figure
RSPCK clock cycle*1	Master	t_{SPcyc}	2	—	4096	t_{Pcyc}	Figure 33.49
	Slave		8	—	4096		
RSPCK clock cycle 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 cycle 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*2	Master	t_{SPCKR}, t_{SPCKF}	—	—	5	ns	
	Slave		—	—	1	t_{Pcyc}	
Data input setup time	Master	t_{SU}	25	—	—	ns	Figures 33.50 to 33.53
	Slave		$20 - 2 \times t_{Pcyc}$	—	—		
Data input hold time	Master	t_H	0	—	—	ns	
	Slave		$20 + 2 \times t_{Pcyc}$	—	—		
SSL setup time	Master	t_{LEAD}	1	—	8	t_{SPcyc}	
	Slave		4	—	—	t_{Pcyc}	
SSL hold time	Master	t_{LAG}	1	—	8	t_{SPcyc}	
	Slave		4	—	—	t_{Pcyc}	
Data output delay time	Master	t_{OD}	—	—	10	ns	
	Slave		—	—	$3 \times t_{Pcyc} + 15$		
Data output hold time	Master	t_{OH}	0	—	—	ns	
	Slave		0	—	—		
Continuous transmission delay time	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	—	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	
	Slave		$4 \times t_{Pcyc}$	—	—		

Item		Symbol	Min.	Typ.	Max.	Unit	Figure
MOSI, MISO rise/fall time*2	Master	t_{DR}, t_{DF}	—	—	5	ns	Figures 33.50 to 33.53
	Slave		—	—	1	t_{Pcyc}	
SSL rise/fall time	Master	t_{SSLR}, t_{SSLF}	—	—	5	ns	
	Slave		—	—	1	t_{Pcyc}	
Slave access time		t_{SA}	—	—	4	t_{Pcyc}	Figures 33.52, 33.53
Slave output release time		t_{REL}	—	—	3	t_{Pcyc}	

Notes: 1. Set t_{Spcyc} so that its value is at least 80 ns.
2. When open drain output is specified, the above timing is not satisfied.

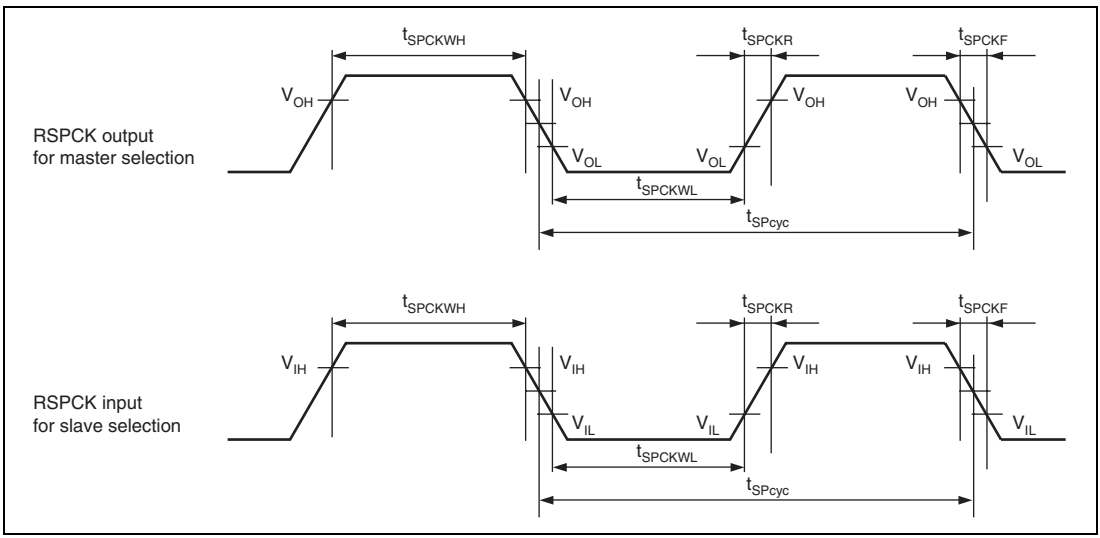


Figure 33.49 SPI Clock Timing

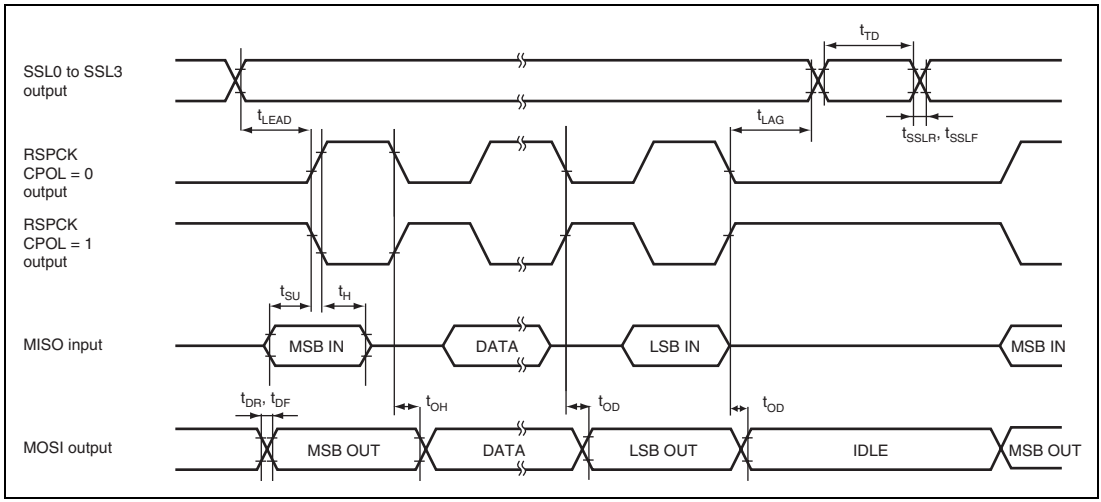


Figure 33.50 SPI Timing (Master, CPHA = 0)

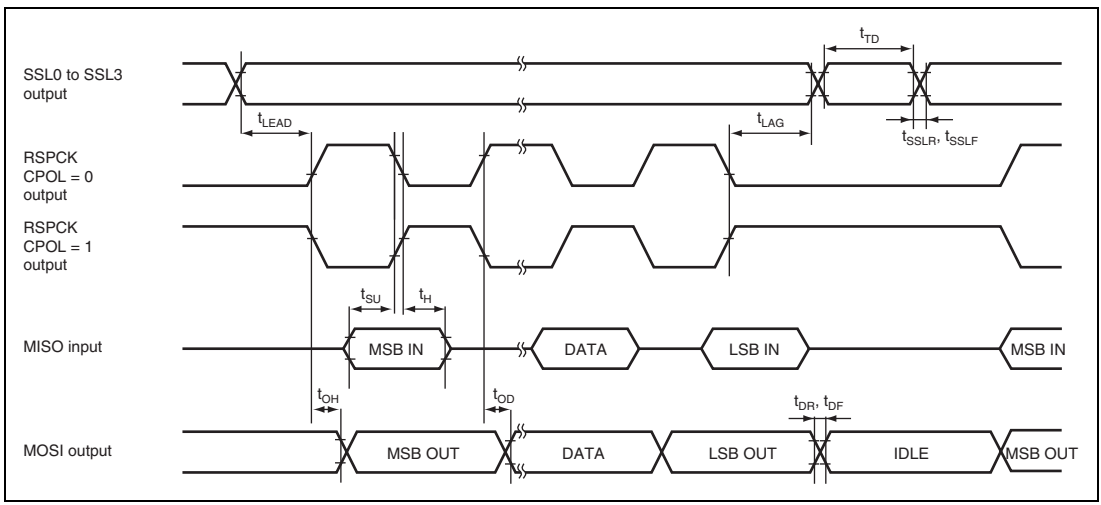


Figure 33.51 SPI Timing (Master, CPHA = 1)

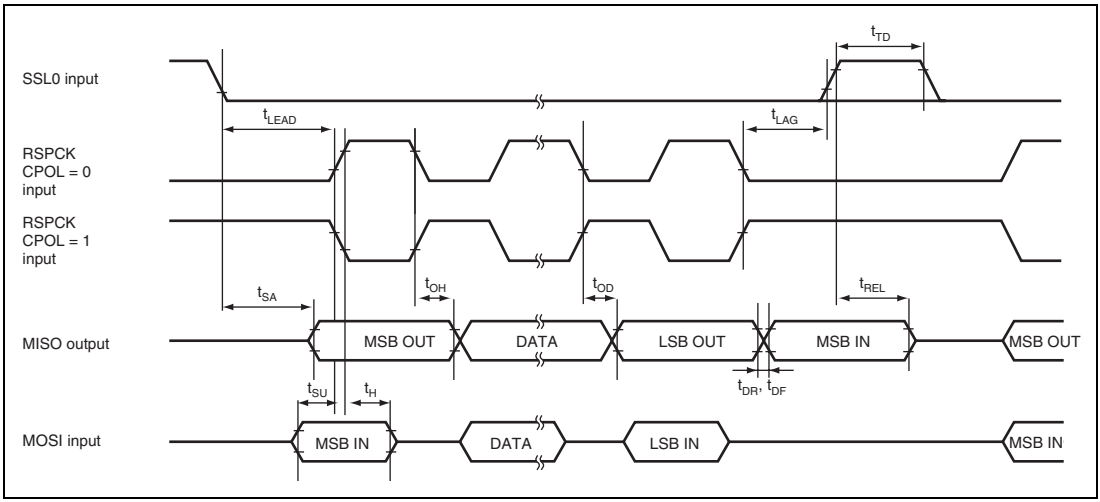


Figure 33.52 SPI Timing (Slave, CPHA = 0)

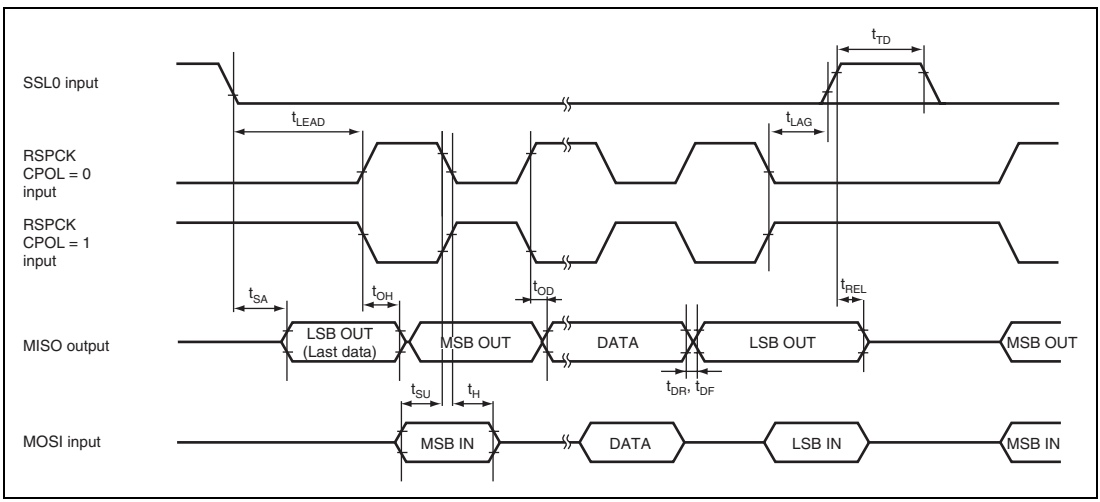


Figure 33.53 SPI Timing (Slave, CPHA = 1)

33.3.13 Controller Area Network (RCAN-ET) Timing

Table 33.17 Controller Area Network (RCAN-ET) Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Transmit data delay time	t_{CTxD}	—	100	ns	Figure 33.54
Receive data setup time	t_{CRxS}	100	—	ns	
Receive data hold time	t_{CRxH}	100	—	ns	

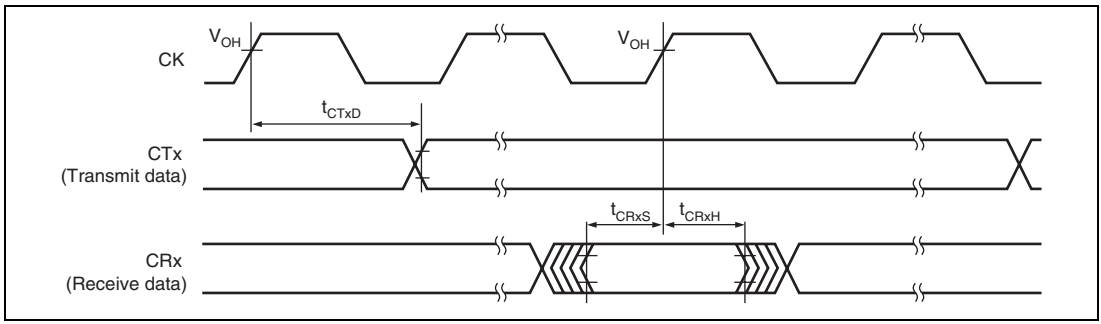


Figure 33.54 RCAN-ET Input/Output Timing

33.3.14 IIC3 Module Timing

Table 33.18 I²C Bus Interface 3 Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

			Specifications				
Item	**Symbol**	**Test Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**	**Figure**
SCL input cycle time	t_{SCL}		12 t_{pcyc} *1 + 600	—	—	ns	Figure 33.55
SCL input high pulse width	t_{SCLH}		3 t_{pcyc} *1 + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		5 t_{pcyc} *1 + 300	—	—	ns	
SCL, SDA input rise time	t_{Sr}		—	—	300	ns	
SCL, SDA input fall time	t_{Sf}		—	—	300	ns	
SCL, SDA input spike pulse removal time*2	t_{SP}		—	—	1 t_{pcyc} *1	ns	
SDA input bus free time	t_{BUF}		5	—	—	t_{pcyc} *1	
Start condition input hold time	t_{STAH}		3	—	—	t_{pcyc} *1	
Retransmit start condition input setup time	t_{STAS}		3	—	—	t_{pcyc} *1	
Stop condition input setup time	t_{STOS}		3	—	—	t_{pcyc} *1	
Data input setup time	t_{SDAS}		1 t_{pcyc} *1 + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA capacitive load	Cb		0	—	400	pF	
SCL, SDA output fall time*3	t_{Sf}		20 + 0.1 Cb	—	250	ns	

Notes: 1. t_{pcyc} indicates peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

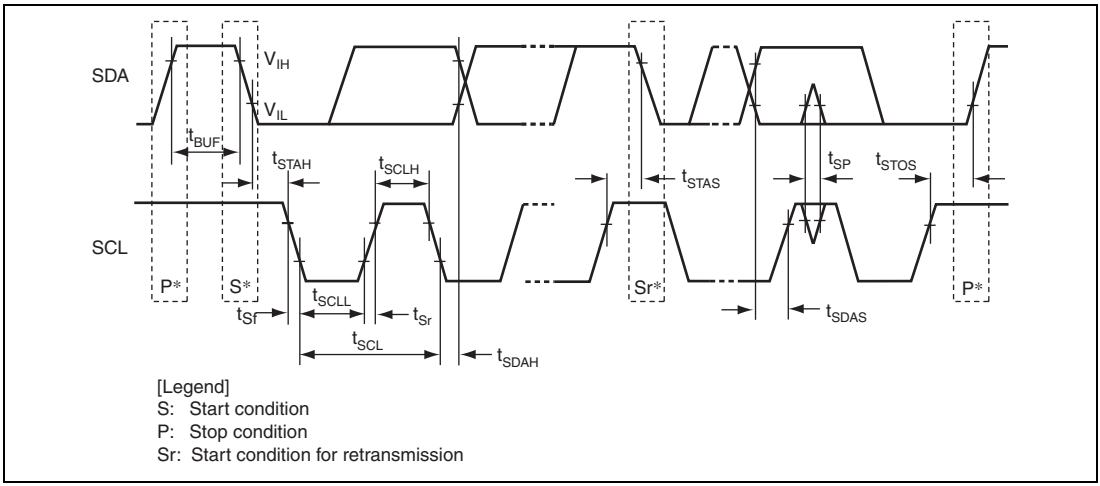


Figure 33.55 I^2C Bus Interface 3 Input/Output Timing

33.3.15 A/D Trigger Input Timing

Table 33.19 A/D Trigger Input Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Module	Item		Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	B:P clock ratio = 1:1	t_{TRGS}	20	—	ns	Figure 33.56
		B:P clock ratio = 2:1		t_{cyc} + 20	—		
		B:P clock ratio = 4:1		$3 \times t_{cyc}$ + 20	—		

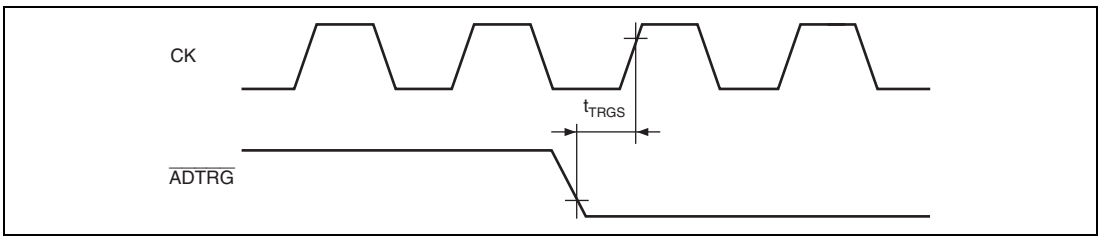


Figure 33.56 A/D Converter External Trigger Input Timing

33.3.16 I/O Port Timing

Table 33.20 I/O Port Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	50	ns	Figure 33.57
Input data setup time	t_{PORTS}	20	—		
Input data hold time	t_{PORTH}	20	—		

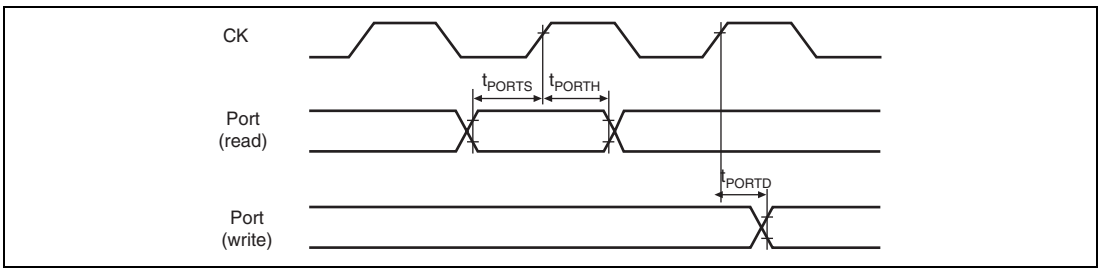


Figure 33.57 I/O Port Timing

33.3.17 EtherC Module Signal Timing

Table 33.21 EtherC Module Signal Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
TX-CLK cycle time	t_{Tcyc}	40	—	ns	—
TX-EN output delay time	t_{TENd}	1	25	ns	Figure 33.58
MII_TXD[3:0] output delay time	t_{MTDd}	1	25	ns	
CRS setup time	t_{CRSs}	10	—	ns	
CRS hold time	t_{CRSh}	10	—	ns	
COL setup time	t_{COLs}	10	—	ns	Figure 33.59
COL hold time	t_{COLh}	10	—	ns	
RX-CLK cycle time	t_{Rcyc}	40	—	ns	—
RX-DV setup time	t_{RDVs}	10	—	ns	Figure 33.60
RX-DV hold time	t_{RDVh}	10	—	ns	
MII_RXD[3:0] setup time	t_{MRDs}	10	—	ns	
MII_RXD[3:0] hold time	t_{MRDh}	10	—	ns	
RX-ER setup time	t_{RERs}	10	—	ns	Figure 33.61
RX-ER hold time	t_{RERh}	10	—	ns	
MDIO setup time	t_{MDIOs}	10	—	ns	Figure 33.62
MDIO hold time	t_{MDIOh}	10	—	ns	
MDIO output data hold time*	t_{MDIOdh}	5	18	ns	Figure 33.63
WOL output delay time	t_{WOLd}	1	25	ns	Figure 33.64
EXOUT output delay time	t_{EXOUTd}	1	20	ns	Figure 33.65

Note: * Users of this LSI need to write and execute a program to make settings that satisfy the above specification.

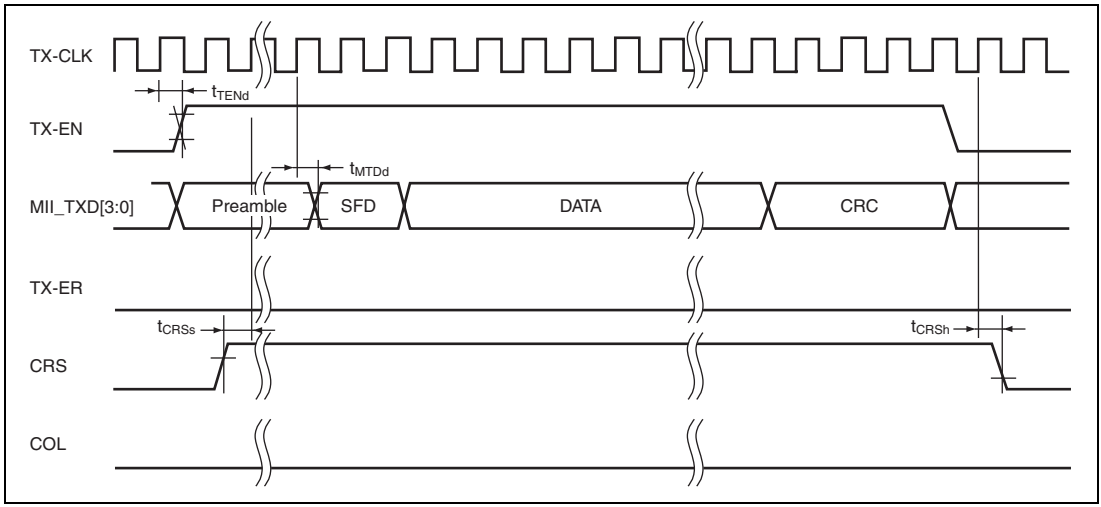


Figure 33.58 MII Transmission Timing (during Normal Operation)

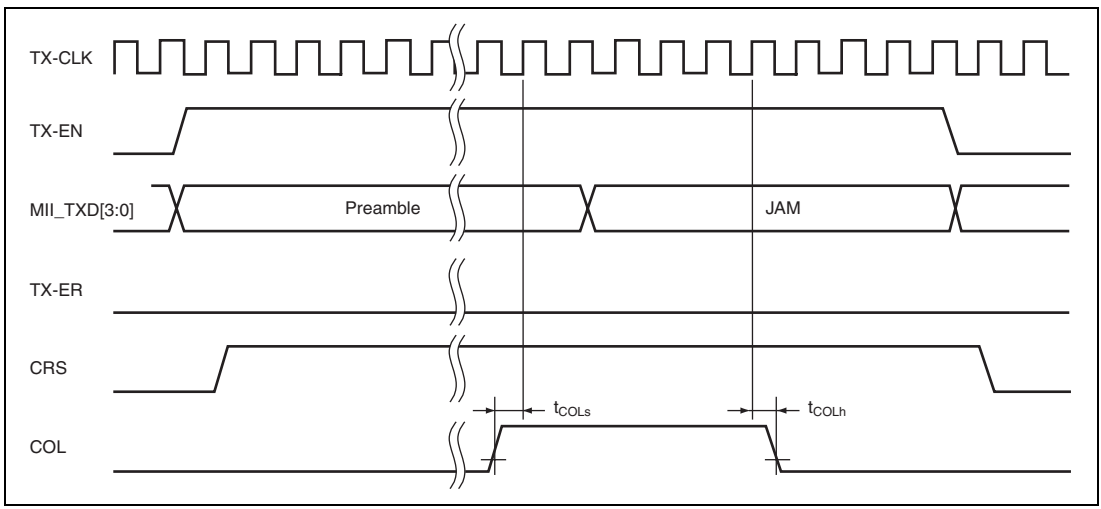


Figure 33.59 MII Transmission Timing (in the Event of a Collision)

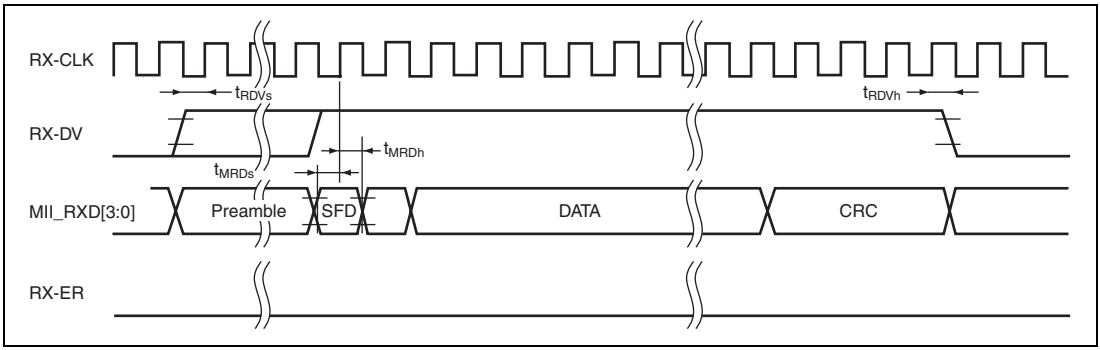


Figure 33.60 MII Reception Timing (during Normal Operation)

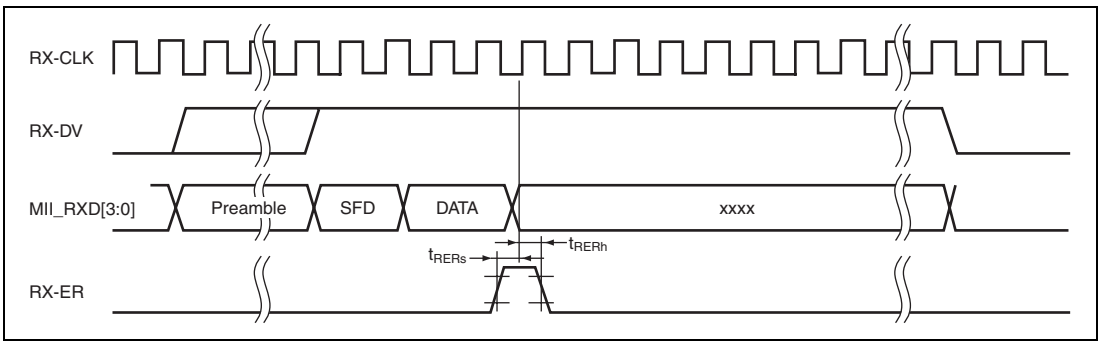


Figure 33.61 MII Reception Timing (in the Event of an Error)

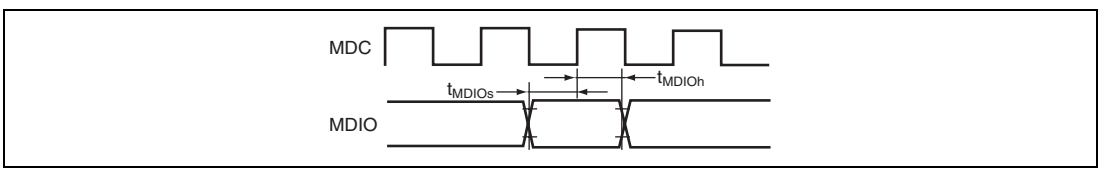


Figure 33.62 MDIO Input Timing

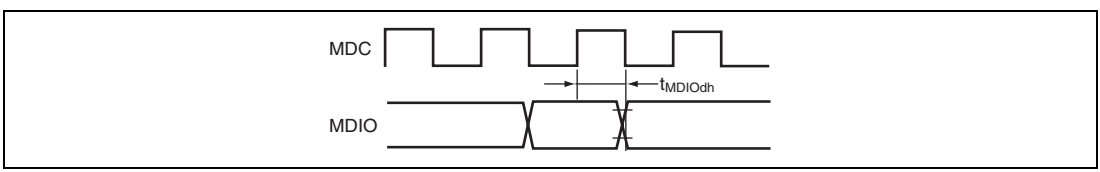


Figure 33.63 MDIO Output Timing

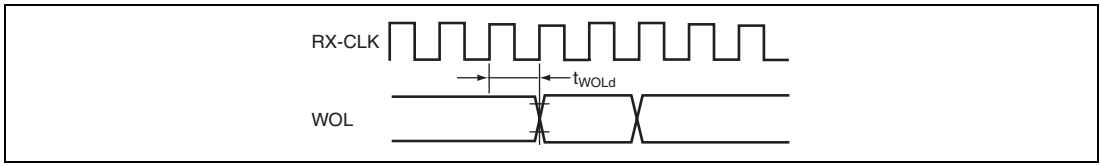


Figure 33.64 WOL Output Timing

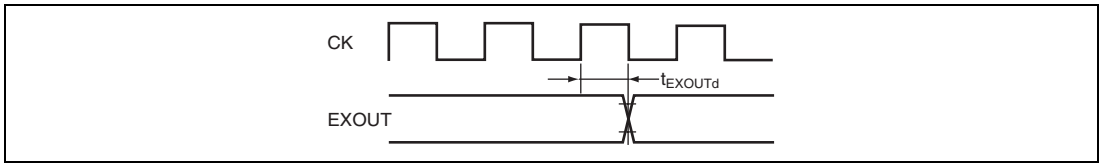


Figure 33.65 EXOUT Output Timing

33.3.18 H-UDI Related Pin Timing

Table 33.22 H-UDI Related Pin Timing

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	40*[1]	—	ns	Figure 33.66
		160*[2]	—	ns	
TCK high pulse width	t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK low pulse width	t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI setup time	t_{TDIS}	15	—	ns	Figure 33.67
TDI hold time	t_{TDIH}	15	—	ns	
TMS setup time	t_{TMSS}	15	—	ns	
TMS hold time	t_{TMSH}	15	—	ns	
TDO delay time	t_{TDOD}	—	30*[1]	ns	
		—	80*[2]	ns	
Output pins other than TDO	t_{OTHERD}	—	80*[2]	ns	

Notes: 1. This value must exceed the cycle time for the peripheral clock (Pϕ).
2. TCK cycle time when the boundary scan function is executed.

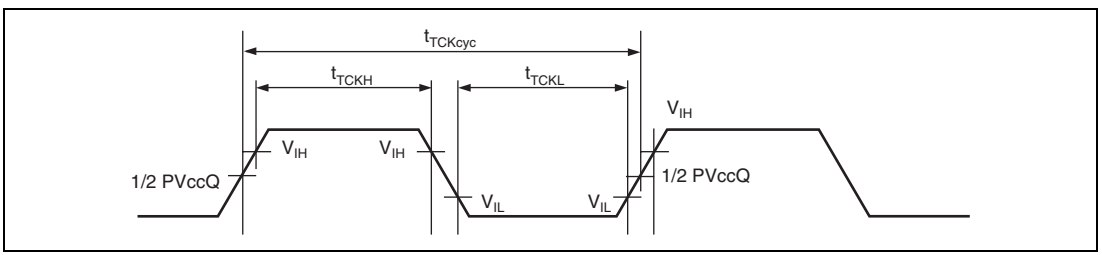


Figure 33.66 TCK Input Timing

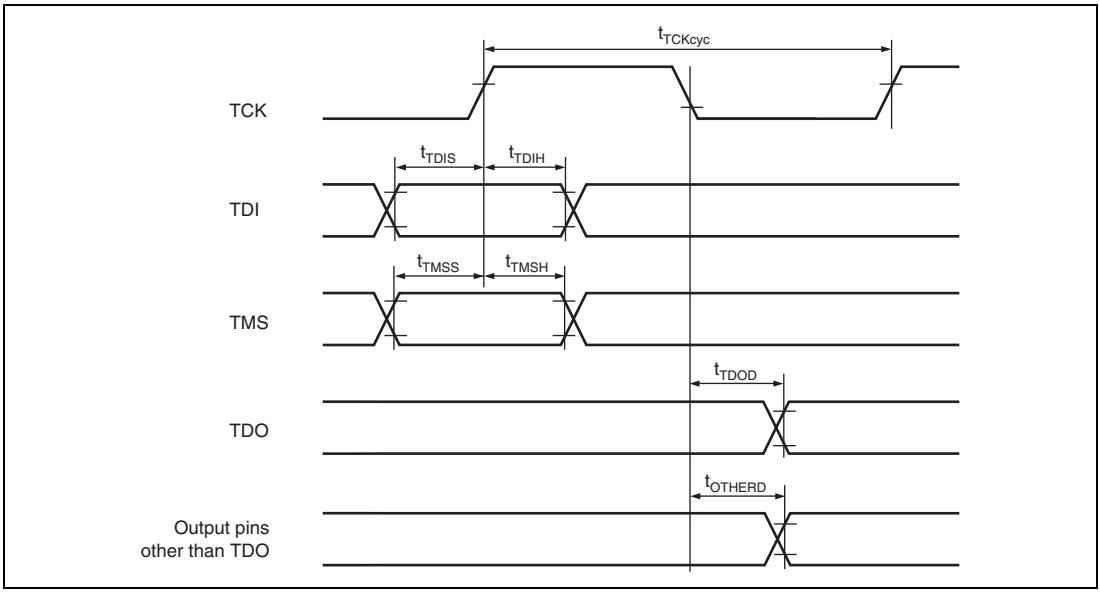
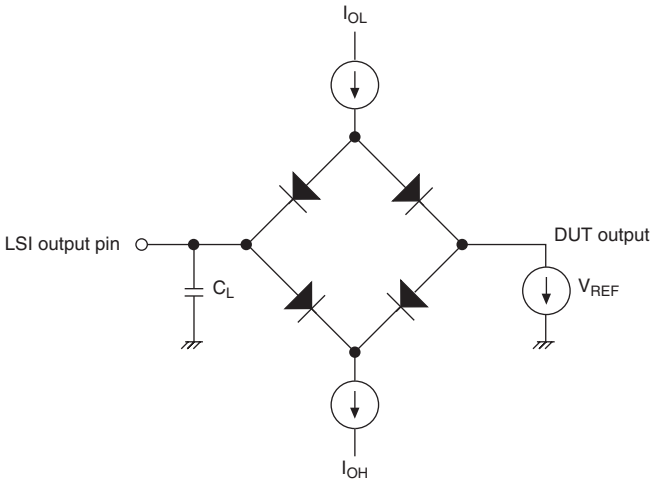


Figure 33.67 H-UDI Data Transmission Timing

33.3.19 AC Characteristics Measurement Conditions

- I/O signal level: V_{IL} (Max.)/V_{IH} (Min.)
- Output signal reference level: High level = 2.0 V, low level = 0.8 V
- Input rise and fall times: 1 ns



Notes: 1. C_L is the total value that includes the capacitance of measurement tools. Each pin is set as follows:
30 pF: CK
30 pF: All pins
2. Test conditions include I_{OL} = 1.6 mA and I_{OH} = –200 μA.

Figure 33.68 Output Load Circuit

33.4 A/D Converter Characteristics

Table 33.23 A/D Converter Characteristics

Conditions: $V_{cc}Q = PLLV_{cc} = DrV_{cc} = 3.0$ to 3.6 V, $AV_{cc} = AVREF = 4.5$ to 5.5V,
$V_{ss} = PLLV_{ss} = DrV_{ss} = AVREFVSS = AV_{ss} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item		Min.	Typ.	Max.	Unit	Test Condition
Resolution		—	12.0	—	bits	
Conversion time		1.0	—	—	µs	Sample & hold circuits or offset cancel circuit is not in use
		1.6	—	—	µs	Sample & hold circuit is in use
Analog input capacitance		—	—	5.0	pF	
Permissible signal-source impedance		—	—	3.0	kΩ	
Nonlinearity error (integral error)		—	—	±4.0	LSB	
Offset error		—	—	±7.5	LSB	
Full-scale error		—	—	±7.5	LSB	
Quantization error		—	—	0.5	LSB	
Absolute accuracy	Sample & hold circuits are in use	—	—	±8.0	LSB	AVin = AVREFVSS + 0.25 V to AVREF – 0.25 V
	Sample & hold circuits are not in use	—	—	±8.0	LSB	AVin = AVREFVSS to AVREF

33.5 USB Characteristics

Table 33.24 USB Characteristics (USD+ and USD- Pins)

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = DrV_{SS} = 0$ V,
Ta = –40°C to +85°C (Industrial specifications)

Item		Symbol	Specifications Min.	Max.	Unit	Test Condition	Figure
Input characteristics	Input high level voltage	V_{IH}	2.0	—	V		Figures 33.69 and 33.70
	Input low level voltage	V_{IL}	—	0.8	V		
	Differential input sense	V_{DI}	0.2	—	V	I(D+) – (D –)I DrV_{CC}* = 3.3 to 3.6 V	
	Differential common mode range	V_{CM}	0.8	2.5	V		
Output characteristics	Output high level voltage	V_{OH}	2.8	—	V	R_L of 15 kΩ to V_{SS}	
	Output low level voltage	V_{OL}	—	0.3	V	R_L of 1.5 kΩ to 3.6 V	
	Crossover voltage	V_{CRS}	1.3	2.0	V		
	Rise time	t_R	4	20	ns		
	Fall time	t_F	4	20	ns		
	Rise time/fall time matching	t_{RFM}	90	111.11	%	(t_R/t_F)	
	Output resistance	Z_{DRV}	28	44	Ω	Including Rs = 22 Ω	

Note: * Be sure to supply the DrV_{CC} with the same voltage as the $V_{CC}Q$.

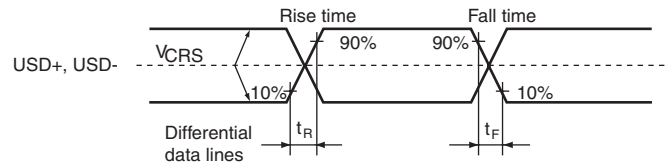


Figure 33.69 Data Signal Timing

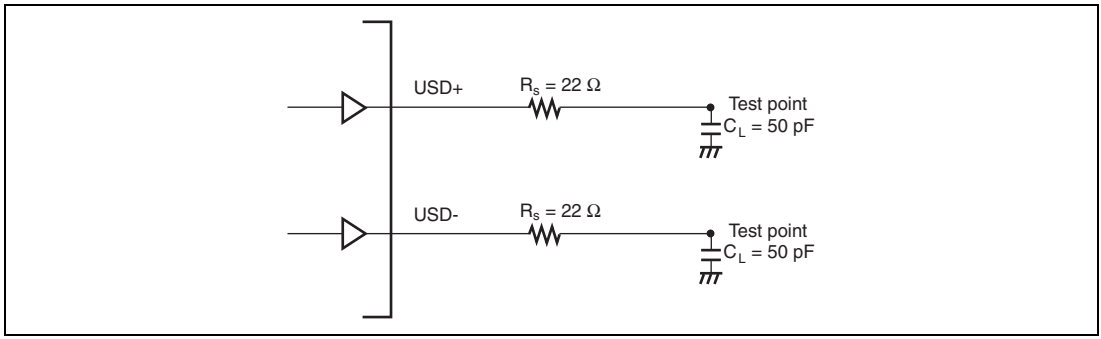


Figure 33.70 Test Load Circuit

33.6 Flash Memory Characteristics

Table 33.25 ROM (Flash Memory for Code Storage) Characteristics

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Operating temperature range during programming/erasing:
Ta = –40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Figure
Programming time	256 bytes	t_{P256}	—	2	12	ms	Pϕ = 50 MHz, $N_{PEC} \leq 100$	
	8 Kbytes	t_{P8K}	—	45	100	ms		
	256 bytes	t_{P256}	—	2.4	14.4	ms	Pϕ = 50 MHz, $N_{PEC} > 100$	
	8 Kbytes	t_{P8K}	—	54	120	ms		
Erase time	8 Kbytes	t_{E8K}	—	50	120	ms	Pϕ = 50 MHz, $N_{PEC} \leq 100$	
	64 Kbytes	t_{E64K}	—	400	875	ms		
	128 Kbytes	t_{E128K}	—	800	1750	ms		
	8 Kbytes	t_{E8K}	—	60	144	ms	Pϕ = 50 MHz, $N_{PEC} > 100$	
	64 Kbytes	t_{E64K}		480	1050	ms		
	128 Kbytes	t_{E128K}		960	2100	ms		
Rewrite/erase cycle*1		N_{PEC}	1000*2	—	—	Times		
Suspend delay time during writing		t_{SPD}	—	—	225	µs	Pϕ = 20 MHz	Figure 33.71
			—	—	175	µs	Pϕ = 40 MHz	
			—	—	155	µs	Pϕ = 50 MHz	
First suspend delay time during erasing (in suspension priority mode)		t_{SESD1}	—	—	220	µs	Pϕ = 20 MHz	
			—	—	130	µs	Pϕ = 40 MHz	
			—	—	120	µs	Pϕ = 50 MHz	
Second suspend delay time during erasing (in suspension priority mode)		t_{SESD2}	—	—	1.7	ms	Pϕ = 50 MHz	
Suspend delay time during erasing (in erasure priority mode)		t_{SEED}	—	—	1.7	ms		
Resume command interval time		t_{RESI}	1.7	—	—	ms		
Data hold time*3		t_{DDRP}	10	—	—	Years		

Notes: 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 32 times for different addresses in 8-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).
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.)
3. This indicates the characteristic when rewrite is performed within the specification range including the minimum number.

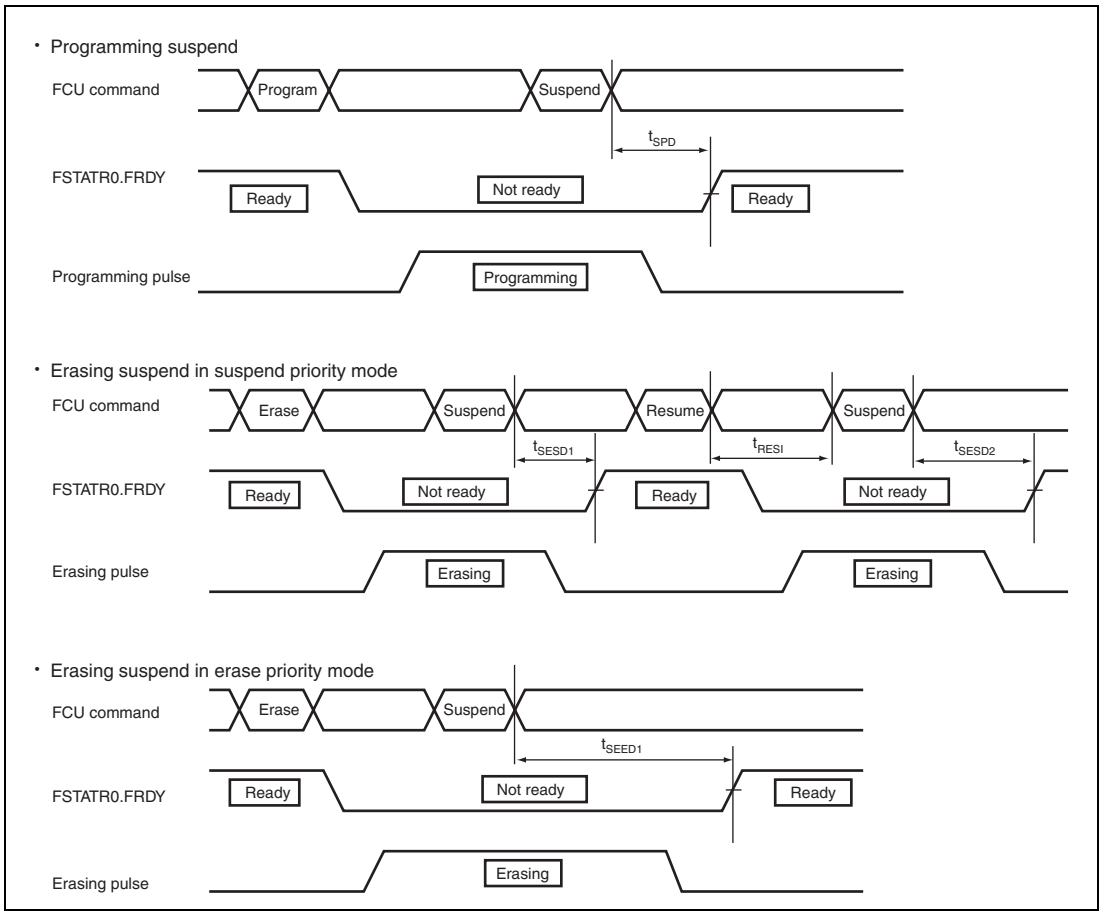


Figure 33.71 Flash Programming/Erasing Suspend Timing

33.7 FLD Characteristics

Table 33.26 FLD (Flash Memory for Data Storage) Characteristics

Conditions: $V_{CC}Q = PLLV_{CC} = DrV_{CC} = 3.0$ to 3.6 V, $AV_{CC} = AVREF = 4.5$ to 5.5V,
$V_{SS} = PLLV_{SS} = DrV_{SS} = AVREFVSS = AV_{SS} = 0$ V,
Operating temperature range during programming/erasing:
Ta = –40°C to +85°C (Industrial specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions	Figure
Programming time	8 bytes	t_{P8}	—	0.4	2	ms	Pϕ = 50 MHz	
	128 bytes	t_{P128}	—	1	5	ms		
Erasure time	8 Kbytes	t_{E8K}	—	300	900	ms	Pϕ = 50 MHz	
Blank check time	8 bytes	t_{BC8}	—	—	30	µs	Pϕ = 50 MHz	
	8 Kbytes	t_{BC8K}	—	—	2.5	ms		
Rewrite/erase cycle*1		N_{PEC}	30000*2	—	—	Times		
Suspend delay time during writing		t_{SPD}	—	—	225	µs	Pϕ = 20 MHz	Figure 33.71
			—	—	175	µs	Pϕ = 40 MHz	
			—	—	155	µs	Pϕ = 50 MHz	
First suspend delay time during erasing (in suspension priority mode)		t_{SESD1}	—	—	220	µs	Pϕ = 20 MHz	
			—	—	130	µs	Pϕ = 40 MHz	
			—	—	120	µs	Pϕ = 50 MHz	
Second suspend delay time during erasing (in suspension priority mode)		t_{SESD2}	—	—	1.7	ms	Pϕ = 50 MHz	
Suspend delay time during erasing in erasure priority mode		t_{SEED}	—	—	1.7	ms		
Resume command interval time		t_{RESI}	1.7	—	—	ms		
Data hold time*3		t_{DDRP}	10	—	—	Years		

Notes: 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 64 times for different addresses in 8-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).

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.)

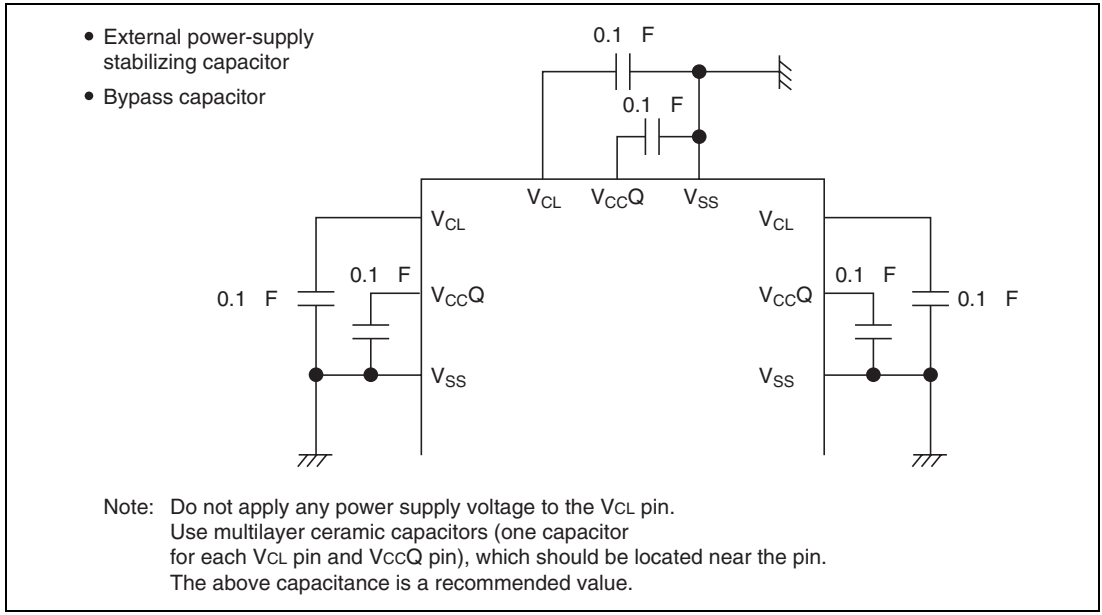
3. This indicates the characteristic when rewrite is performed within the specification range including the minimum number.

33.8 Usage Notes

33.8.1 Notes on Connecting Capacitors

This LSI includes an internal step-down circuit to automatically reduce the internal power supply voltage to an appropriate level. Between this internal stepped-down power supply (V_{CL} pin) and the V_{SS} pin, a capacitor for stabilizing the internal voltage needs to be connected. Connection of the external capacitor is shown in figure 33.72. The external capacitor should be located near the pin. Do not apply any power supply voltage to the V_{CL} pin.

A multilayer ceramic capacitor should be inserted for each pair of power supply pins as a bypass capacitor. The bypass capacitor must be inserted as close to the power supply pins of the LSI as possible. Connect the bypass capacitor and the capacitor for stabilizing the internal voltage with the capacitance from 0.02 to 0.33 µF, after being evaluated in the system. For details on capacitors related to crystal oscillation, see section 4.9, Notes on Board Design.



Note: Do not apply any power supply voltage to the VCL pin.
Use multilayer ceramic capacitors (one capacitor for each VCL pin and VCCQ pin), which should be located near the pin.
The above capacitance is a recommended value.

Figure 33.72 Connection of Capacitors

Appendix

A. Pin States

Pin initial states differ according to MCU operating modes. Refer to section 22, Pin Function Controller (PFC), for details.

Table A.1 Pin States

Pin Function		Pin State										
		Reset State					Power-Down State					
		Power-On				Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
		Expansion without ROM		Expansion with ROM	Single Chip						
Type	Pin Name	16 Bits	32 Bits								
Clock	CK	O			Z	O	Z*4	O	Z*4	O	O
	XTAL	O				O	L	O	O	O	O
	EXTAL	I				I	I	I	I	I	I
System control	$\overline{RES}$	I				I	I	I	I	I	I
	$\overline{MRES}$	Z				I	I*7	I	I	I*7	I
	$\overline{WDTOVF}$	O*9				O	O	O	O	O	O
	$\overline{BREQ}$	Z				I	Z	I	I	I	I
	$\overline{BACK}$	Z				O	Z	O	L	O	O
Operating mode control	MD0, MD1	I				I	I	I	I	I	I
	$\overline{ASEMD0}$	I*10				I*10	I*10	I*10	I*10	I*10	I*10
	FWE	I				I	I	I	I	I	I
Interrupt	NMI	I				I	I	I	I	I	I
	IRQ0 to IRQ7	Z				I	I	I	I	I	I
	$\overline{IRQOUT}$ (PE15)	Z				O	Z (MZIZEH in HCPCR = 0)	O	O	O*7	O
							H*1 (MZIZEH in HCPCR = 1)				
	$\overline{IRQOUT}$ (PE1)	Z				O	H*1	O	O	O	O

Pin Function		Pin State									
		Reset State					Power-Down State				
		Power-On									
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
Address bus	A0 to A25	O		Z		O	Z*3	O	Z	O	O
Data bus	D0 to D9, D16 to D23, D30, D31	Z				I/O	Z	I/O	Z	I/O	I/O
	D10 to D15	Z				I/O	Z	I/O	Z	I/O*6	I/O
	D24 to D29	Z				I/O	Z	I/O	Z	I/O*5	I/O
Bus control	$\overline{\text{WAIT}}$	Z				I	Z	I	Z	I	I
	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	H		Z		O	Z*3	O	Z	O	O
	$\overline{\text{CS2}}$ to $\overline{\text{CS7}}$	Z				O	Z*3	O	Z	O	O
	$\overline{\text{BS}}$	Z				O	Z*3	O	Z	O	O
	$\overline{\text{RASU}}$, $\overline{\text{RASL}}$	Z				O	Z*2	O	Z*2	O	O
	$\overline{\text{CASU}}$, $\overline{\text{CASL}}$	Z				O	Z*2	O	Z*2	O	O
	DQMUU, DQMUL, DQMLU, DQMLL	Z				O	Z*3	O	Z	O	O
	$\overline{\text{AH}}$	Z				O	Z*3	O	Z	O	O
	$\overline{\text{FRAME}}$	Z				O	Z*3	O	Z	O	O
	RD/$\overline{\text{WR}}$	Z				O	Z*3	O	Z	O	O
	$\overline{\text{RD}}$	H		Z		O	Z*3	O	Z	O	O
	$\overline{\text{WRHH}}$, $\overline{\text{WRHL}}$	Z	H	Z		O	Z*3	O	Z	O	O
	$\overline{\text{WRH}}$, $\overline{\text{WRL}}$	H		Z		O	Z*3	O	Z	O	O
	CKE	Z				O	Z*2	O	Z*2	O	O
	$\overline{\text{REFOUT}}$ (PE15)	Z				O	Z (MZIZEH in HCPCR = 0)	O	O	O*7	O
							H*1 (MZIZEH in HCPCR = 1)				
	$\overline{\text{REFOUT}}$ (PE1)	Z				O	H*1	O	O	O	O

Pin Function		Pin State										
		Reset State					Power-Down State					
		Power-On										
		Expansion without ROM										
Type	Pin Name	16 Bits	32 Bits	Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
DMAC	DREQ0 (PE0), DREQ1 (PE2)	Z				I	Z	I	I	I*8	I
	DREQ0 (PB8), DREQ1 (PD22), DREQ2, DREQ3	Z				I	Z	I	I	I	I
	DACK0 (PE14), DACK1 (PE15), DACK2, DACK3	Z				O	Z (MZIZEH in HCPCR = 0) O*1 (MZIZEH in HCPCR = 1)	O	O	O*7	O
	DACK0 (PB9), DACK1 (PD23)	Z				O	O*1	O	O	O	O
	TEND0 (PE1), TEND1 (PE3)	Z				O	Z (MZIZEL in HCPCR = 0) O*1 (MZIZEL in HCPCR = 1)	O	O	O*8	O
	TEND0 (PB7), TEND1 (PD21)	Z				O	O*1	O	O	O	O
MTU2	TCLKA to TCLKD	Z				I	Z	I	I	I	I
	TIOC0A (PE0), TIOC0B (PE1), TIOC0C (PE2), TIOC0D (PE3)	Z				I/O	Z (MZIZEL in HCPCR = 0) K*1 (MZIZEL in HCPCR = 1)	I/O	I/O	I/O*8	Z
	TIOC0A (PB1), TIOC0B (PB2), TIOC0C (PB3), TIOC0D (PB4)	Z				I/O	K*1	I/O	I/O	I/O	Z
	TIOC1A	Z				I/O	K*1	I/O	I/O	I/O	I/O

Pin Function		Pin State										
		Reset State					Power-Down State					
		Power-On										
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
MTU2	TIOC1B (PE5), TIOC2A (PE6)	Z				I/O	Z (MZIZEL in HCPCR = 0) K*1 (MZIZEL in HCPCR = 1)	I/O	I/O	I/O*8	I/O
	TIOC1B (PC11), TIOC2A (PB0)	Z				I/O	K*1	I/O	I/O	I/O	I/O
	TIOC2B	Z				I/O	K*1	I/O	I/O	I/O	I/O
	TIOC3A, TIOC3C	Z				I/O	K*1	I/O	I/O	I/O	I/O
	TIOC3B, TIOC3D	Z				I/O	Z (MZIZEH in HCPCR = 0) K*1 (MZIZEH in HCPCR = 1)	I/O	I/O	I/O*7	Z
	TIOC4A, TIOC4B, TIOC4C, TIOC4D	Z				I/O	Z (MZIZEH in HCPCR = 0) K*1 (MZIZEH in HCPCR = 1)	I/O	I/O	I/O*7	Z
	TIC5U, TIC5V, TIC5W	Z				I	Z	I	I	I	I
MTU2S	TIOC3AS, TIOC3CS	Z				I/O	K*1	I/O	I/O	I/O	I/O
	TIOC3BS (PD10), TIOC3DS (PD11), TIOC4AS (PD12), TIOC4BS (PD13), TIOC4CS (PD14), TIOC4DS (PD15)	Z				I/O	Z (MZIZDL in HCPCR = 0) K*1 (MZIZDL in HCPCR = 1)	I/O	I/O	I/O*6	Z

Pin Function		Pin State										
		Reset State					Power-Down State					
		Power-On										
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
MTU2S	TIOC3BS (PD29), TIOC3DS (PD28), TIOC4AS (PD27), TIOC4BS (PD26), TIOC4CS (PD25), TIOC4DS (PD24)	Z				I/O	Z (MZIZDH in HCPCR = 0)	I/O	I/O	I/O*5	Z
							K*1 (MZIZDH in HCPCR = 1)				
	TIOC3BS (PE5), TIOC3DS (PE6), TIOC4AS (PE0), TIOC4BS (PE1), TIOC4CS (PE2), TIOC4DS (PE3)	Z				I/O	Z (MZIZEL in HCPCR = 0)	I/O	I/O	I/O*8	Z
							K*1 (MZIZEL in HCPCR = 1)				
	TIC5US, TIC5VS, TIC5WS	Z				I	Z	I	I	I	I
POE2	$\overline{\text{POE0}}$ to $\overline{\text{POE4}}$, $\overline{\text{POE8}}$	Z				I	Z	I	I	I	I
SCI	SCK0 to SCK2, SCK4	Z				I/O	K*1	I/O	I/O	I/O	I/O
	RXD0 to RXD2, RXD4	Z				I	Z	I	I	I	I
	TXD0 to TXD2, TXD4	Z				O	O*1	O	O	O	O
SCIF	SCK3	Z				I/O	K*1	I/O	I/O	I/O	I/O
	RXD3 (PB2)	Z				I	Z	I	I	I	I
	RXD3 (PE6)	Z				I	Z	I	I	I*8	I
	TXD3 (PE5)	Z				O	Z (MZIZEL in HCPCR = 0)	O	O	O*8	O
							O*1 (MZIZEL in HCPCR = 1)				
	TXD3 (PB3)	Z				O	O*1	O	O	O	O

Pin Function		Pin State									
		Reset State					Power-Down State				
		Power-On									
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
RSPI	RSPCK	Z				I/O	K*1	I/O	I/O	I/O	I/O
	SSL0	Z				I/O	K*1	I/O	I/O	I/O	I/O
	SSL1 to SSL3	Z				O	K*1	O	O	O	O
	MOSI	Z				I/O	Z	I/O	I/O	I/O	I/O
	MISO	Z				I/O	K*1	I/O	I/O	I/O	I/O
IIC3	SCL	Z				I/O	Z	I/O	I/O	I/O	I/O
	SDA	Z				I/O	Z	I/O	I/O	I/O	I/O
UBC	$\overline{\text{UBCTRG}}$	Z				O	O*1	O	O	O	O
A/D converter	AN0 to AN7	Z				I	Z	I	I	I	I
	$\overline{\text{ADTRG}}$	Z				I	Z	I	I	I	I
USB	USBXTAL	O				O	L	O	O	O	O
	USBEXTAL	I				I	I	I	I	I	I
	VBUS	I				I	I	I	I	I	I
	USD+	Z				I/O	I	I/O	I/O	I/O	I/O
	USD-	Z				I/O	I	I/O	I/O	I/O	I/O
RCAN-ET	CRx0	Z				I	Z	I	I	I	I
	CTx0	Z				O	O*1	O	O	O	O
I/O port	PA0 to PA21	Z				I/O	K*1	I/O	I/O	I/O	I/O
	PB0 to PB11, PB14, PB15	Z				I/O	K*1	I/O	I/O	I/O	I/O
	PB12, PB13	Z				I	Z	I	I	I	I
	PC0 to PC15	Z				I/O	K*1	I/O	I/O	I/O	I/O
	PD0 to PD9, PD16 to PD23, PD30, PD31	Z				I/O	K*1	I/O	I/O	I/O	I/O

Pin Function		Pin State										
		Reset State					Power-Down State					
		Power-On										
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
I/O port	PD10 to PD15	Z				I/O	Z (MZIZDL in HCPCR = 0) K*1 (MZIZDL in HCPCR = 1)	I/O	I/O	I/O*6	Z
	PD24 to PD29	Z				I/O	Z (MZIZDH in HCPCR = 0) K*1 (MZIZDH in HCPCR = 1)	I/O	I/O	I/O*5	Z
	PE4, PE7, PE8, PE10	Z				I/O	K*1	I/O	I/O	I/O	I/O
	PE0 to PE3, PE5, PE6	Z				I/O	Z (MZIZEL in HCPCR = 0) K*1 (MZIZEL in HCPCR = 1)	I/O	I/O	I/O*8	Z
	PE9, PE11 to PE15	Z				I/O	Z (MZIZEH in HCPCR = 0) K*1 (MZIZEH in HCPCR = 1)	I/O	I/O	I/O*7	Z
	PF0 to PF7	Z				I	Z	I	I	I	I
Ether	RX_ER	Z				I	I	I	I	I	I
	RX_DV	Z				I	I	I	I	I	I
	TX_CLK	Z				I	I	I	I	I	I
	LNKSTA (PD19)	Z				I	Z	I	I	I	I

Pin Function		Pin State									
		Reset State					Power-Down State				
		Power-On									
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
Ether	COL (PD23), CRS (PE4), RX_CLK (PA0), MII_RXD0 (PA1), MII_RXD1 (PA2), MII_RXD2 (PA3), MII_RXD3 (PA4)	Z				I	I	I	I	I	I
	LNKSTA (PE0)	Z				I	Z	I	I	I*8	I
	COL (PE3)	Z				I	Z (MZIZEL in HCPCR = 0)	I	I	I*8	I
							I (MZIZEL in HCPCR = 1)				
	CRS (PD24), RX_CLK (PD25), MII_RXD0 (PD26), MII_RXD1 (PD27), MII_RXD2 (PD28), MII_RXD3 (PD29)	Z				I	Z (MZIZDH in HCPCR = 0)	I	I	I*5	I
							I (MZIZDH in HCPCR = 1)				
	MDC (PD20), TX_EN (PA11), MII_TXD0 (PA10), MII_TXD1 (PA9), MII_TXD2 (PA8), MII_TXD3 (PA7), TX_ER (PA6)	Z				O	O*1	O	O	O	O
	EXOUT	Z				O	O*1	O	O	O	O
	WOL (PD22)	Z				O	O*1	O	O	O	O

Pin Function		Pin State										
		Reset State					Power-Down State					
		Power-On										
		Expansion without ROM		Expansion with ROM	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release	Oscillation Stop Detected	POE Function Used
Type	Pin Name	16 Bits	32 Bits								
Ether	WOL (PE2)	Z				O	Z (MZIZEL in HCPCR = 0) O*1 (MZIZEL in HCPCR = 1)	O	O	O*8	O
	MDC (PE1)	Z				O	Z (MZIZEL in HCPCR = 0) O*1 (MZIZEL in HCPCR = 1)	O	O	O*8	O
	TX_EN (PE9), MII_TXD0 (PE11), MII_TXD1 (PE12), MII_TXD2 (PE13), MII_TXD3 (PE14), TX_ER (PE15)	Z				O	Z (MZIZEH in HCPCR = 0) O*1 (MZIZEH in HCPCR = 1)	O	O	O*7	O
	MDIO (PD18)	Z				I/O	I/O*1	I/O	I/O	I/O	I/O
	MDIO (PE5)	Z				I/O	Z (MZIZEL in HCPCR = 0) I/O*1 (MZIZEL in HCPCR = 1)	I/O	I/O	I/O*9	I/O

[Legend]

I: Input

O: Output

H: High-level output

L: Low-level output

Z: High-impedance

K: Input pins become high-impedance, and output pins retain their state.

Notes: 1. Output pins become high-impedance when the HIZ bit in standby control register 3 (STBCR3) is set to 1.
2. Becomes output when the HIZCNT bit in the common control register (CMNCR) is set to 1.
3. Becomes output when the HIZMEM bit in the common control register (CMNCR) is set to 1.
4. Becomes output when the HIZCKIO bit in the common control register (CMNCR) is set to 1.
5. Becomes high-impedance when the MZIZDH bit in the high-current port control register (HCPCR) is set to 0.
6. Becomes high-impedance when the MZIZDL bit in the high-current port control register (HCPCR) is set to 0.
7. Becomes high-impedance when the MZIZEH bit in the high-current port control register (HCPCR) is set to 0.
8. Becomes high-impedance when the MZIZEL bit in the high-current port control register (HCPCR) is set to 0.
9. Becomes input during a power-on reset. Pull-up to prevent erroneous operation. Pull-down with a resistance of at least 1 MW as required.
10. Pulled-up inside the LSI when there is no input.

B. Product Code Lineup

Table B.1 Product Code Lineup

Product Type							
Product Name	Classification	ROM Capacity	RAM Capacity	Application	Operating temperature	Product Code	Package
SH7216A	F-ZTAT (FPU and Ether functions enabled, and Iϕ = 200 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72167ADFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72167ADFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72167ADBG	PLBG0176GA-A BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72166ADFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72166ADFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72166ADBG	PLBG0176GA-A BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72165ADFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72165ADFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72165ADBG	PLBG0176GA-A BP-176V
SH7216B	F-ZTAT (FPU function enabled, Ether function disabled, and Iϕ = 200 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72167BDFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72167BDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72167BDBG	PLBG0176GA-A BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72166BDFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72166BDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72166BDBG	PLBG0176GA-A BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72165BDFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72165BDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72165BDBG	PLBG0176GA-A BP-176V

Product Type							
Product Name	**Classification**	**ROM Capacity**	**RAM Capacity**	**Application**	**Operating temperature**	**Product Code**	**Package**
SH7216G	F-ZTAT (FPU and Ether functions enabled, and Iϕ = 100 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72167GDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72167GDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72167GDBG	PLBG0176GA-A
							BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72166GDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72166GDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72166GDBG	PLBG0176GA-A
							BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72165GDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72165GDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72165GDBG	PLBG0176GA-A
							BP-176V
SH7216H	F-ZTAT (FPU function enabled, Ether function disabled, and Iϕ = 100 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72167HDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72167HDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72167HDBG	PLBG0176GA-A
							BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72166HDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72166HDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72166HDBG	PLBG0176GA-A
							BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72165HDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72165HDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72165HDBG	PLBG0176GA-A
							BP-176V

Product Type							
Product Name	**Classification**	**ROM Capacity**	**RAM Capacity**	**Application**	**Operating temperature**	**Product Code**	**Package**
SH7214A	F-ZTAT (FPU function disabled, Ether function enabled, and Iϕ = 200 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72147ADFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72147ADFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72147ADBG	PLBG0176GA-A BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72146ADFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72146ADFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72146ADBG	PLBG0176GA-A BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72145ADFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72145ADFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72145ADBG	PLBG0176GA-A BP-176V
SH7214B	F-ZTAT (FPU and Ether functions disabled, and Iϕ = 200 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72147BDFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72147BDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72147BDBG	PLBG0176GA-A BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72146BDFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72146BDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72146BDBG	PLBG0176GA-A BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72145BDFP	PLQP0176KB-A FP-176EV
				Industrial application	–40 to +85 °C	R5F72145BDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72145BDBG	PLBG0176GA-A BP-176V

Product Type							
Product Name	**Classification**	**ROM Capacity**	**RAM Capacity**	**Application**	**Operating temperature**	**Product Code**	**Package**
SH7214G	F-ZTAT (FPU function disabled, Ether function enabled, and Iϕ = 100 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72147GDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72147GDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72147GDBG	PLBG0176GA-A
							BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72146GDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72146GDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72146GDBG	PLBG0176GA-A
							BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72145GDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72145GDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72145GDBG	PLBG0176GA-A
							BP-176V
SH7214H	F-ZTAT (FPU and Ether functions disabled, and Iϕ = 100 MHz)	1 Mbyte	128 Kbytes	Industrial application	–40 to +85 °C	R5F72147HDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72147HDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72147HDBG	PLBG0176GA-A
							BP-176V
		768 Kbytes	96 Kbytes	Industrial application	–40 to +85 °C	R5F72146HDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72146HDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72146HDBG	PLBG0176GA-A
							BP-176V
		512 Kbytes	64 Kbytes	Industrial application	–40 to +85 °C	R5F72145HDFP	PLQP0176KB-A
							FP-176EV
				Industrial application	–40 to +85 °C	R5F72145HDFA	PLQP0176LA-B
				Industrial application	–40 to +85 °C	R5F72145HDBG	PLBG0176GA-A
							BP-176V

C. Package Dimensions

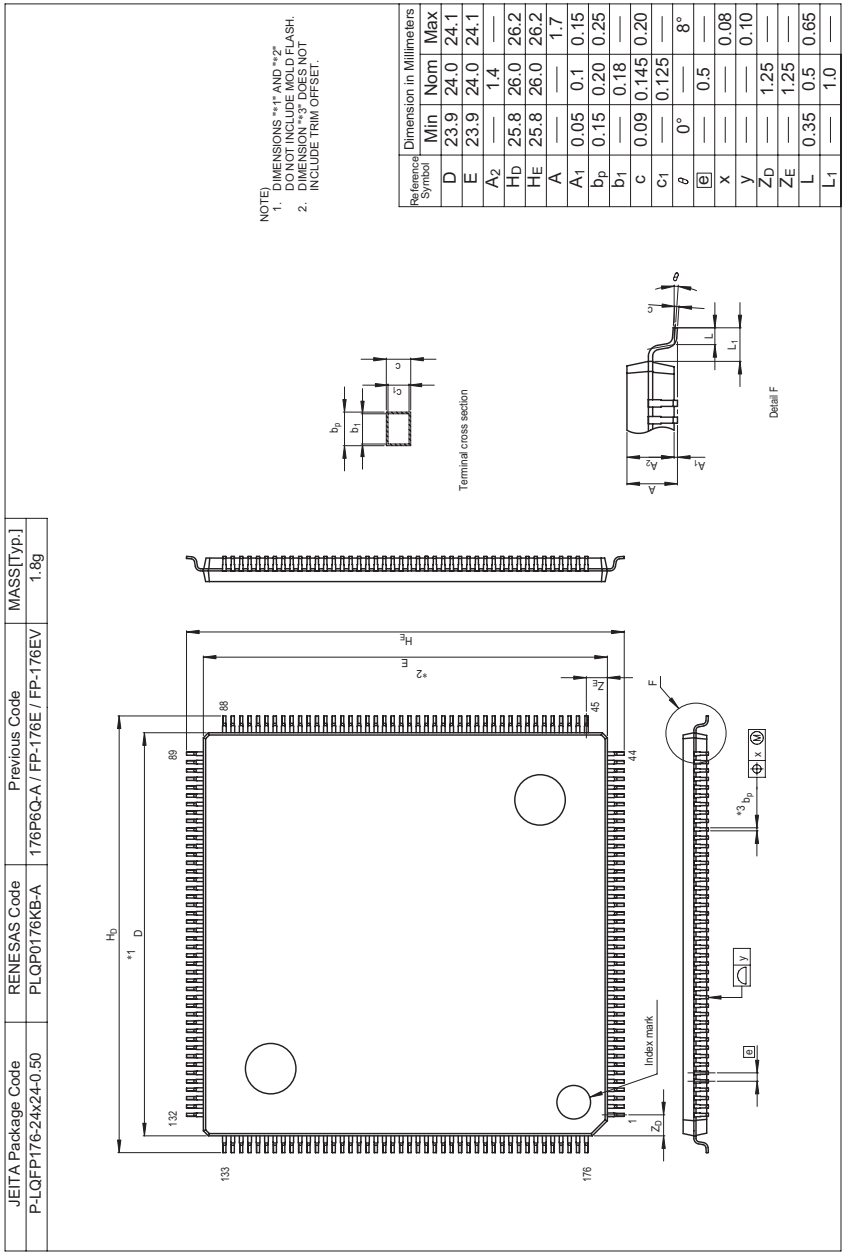


Figure C.1 Package Dimensions (1)

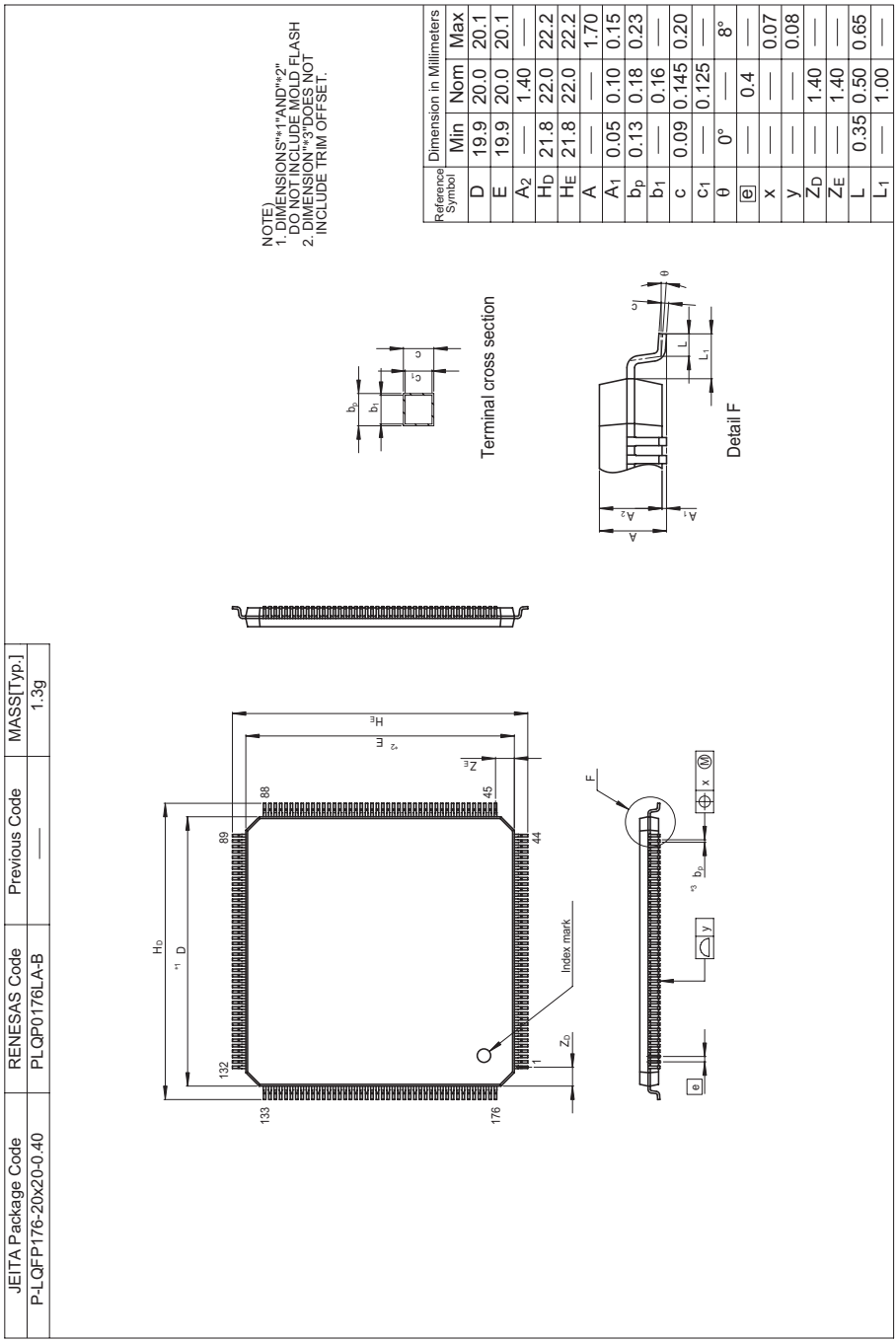
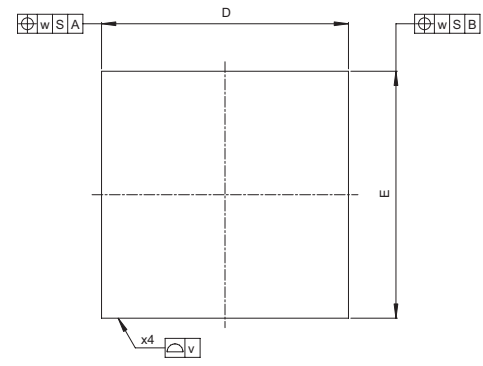
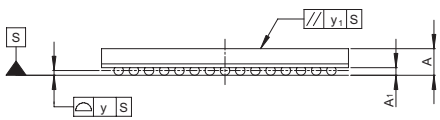
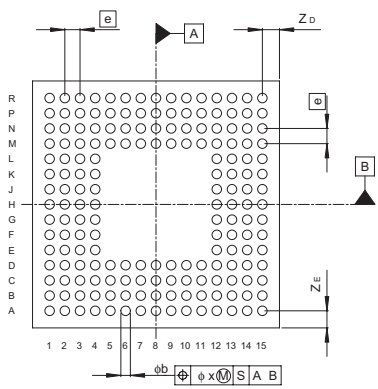


Figure C.2 Package Dimensions (2)

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFBGA176-13x13-0.80	PLBG0176GA-A	BP-176/BP-176V	0.45g



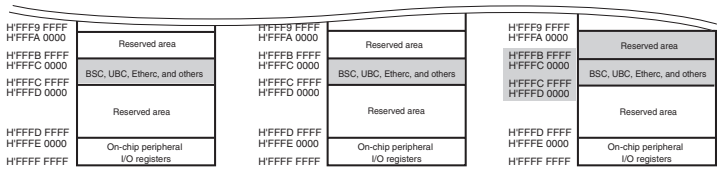
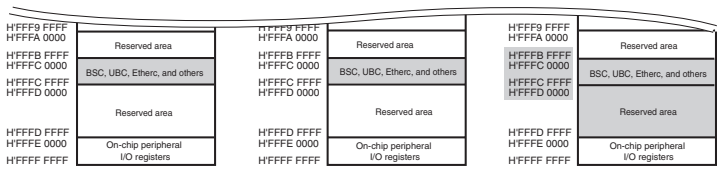
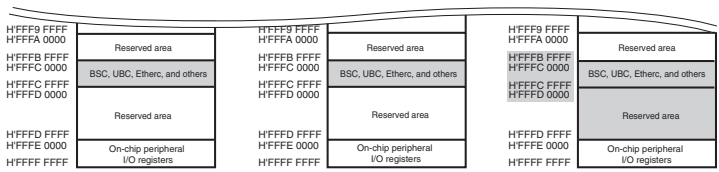




Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Figure C.3 Package Dimensions (3)

Main Revisions and Additions in this Edition

Item	Page	Revision (See Manual for Details)
Table 1.2 Pin Functions	13	Added **Classification** / **Symbol** / **I/O** / **Name** / **Function** System control / $\overline{\text{WDTOVF}}$ / Output / Watchdog timer overflow / Outputs an overflow signal from the WDT. Use a resistor with a value of at least 1 MΩ to pull this pin down.
Figure 3.1 Address Map (1-Mbyte Version)	77	Amended and added 
Figure 3.2 Address Map (768-Kbyte Version)	78	Amended and added 
Figure 3.3 Address Map (512-Kbyte Version)	79	Amended and added 
4.7 Oscillation Stop Detection	99	Deleted In addition, the high-current ports (multiplexed pins to which the TIOC3B, TIOC3D, and TIOC4A to TIOC4D signals in the MTU2, the TIOC3BS, TIOC3DS, and TIOC4AS to TIOC4DS in the MTU2S are assigned) can be placed in high-impedance state regardless of settings of the OSCERS bit and PFC. ~~For details, refer to appendix A, Pin Status.~~

Item	Page	Revision (See Manual for Details)
8.9.11 Note on USB as DTC Activation Sources	251	Amended To generate a CPU interrupt when a DTC transfer activated by the USB is completed, refer to the procedure described in section 24, USB Function Module (USB).
Table 9.2 Address Map in On-Chip ROM-Enabled Mode	258	Added

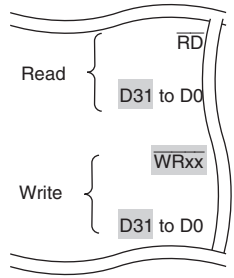
Memory to be Connected	Size
On-chip ROM	512 kbytes (SH72165,SH72145), 768 kbytes (SH72166, SH72146), 1 Mbyte (SH72167, SH72147)

9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7) — 282 — Amended and added

(1) Normal Space, SRAM with Byte Selection, MPX-I/O

- CS5WCR

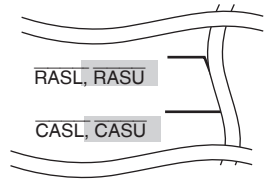
Bit	Bit Name	Description
12, 11	SW[1:0]	Number of Delay Cycles from Address, CS5 Assertion to RD, WRxx Assertion Specify the number of delay cycles from address and CS5 assertion to RD and WRxx assertion when area 5 is specified as normal space or SRAM with byte selection. Specify the number of delay cycles from the end of address cycle (Ta3) to RD and WRxx assertion when area 5 is specified as MPx-I/O. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
1, 0	HW[1:0]	Delay Cycles from RD, WRxx Negation to Address, CS5 Negation Specify the number of delay cycles from RD and WRxx negation to address and CS5 negation when area 5 is specified as normal space or SRAM with byte selection. Specify the number of delay cycles from RD and WRxx negation to CS5 negation when area 5 is specified as MPx-I/O. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Item	Page	Revision (See Manual for Details)
Figure 9.8 Wait Timing for Normal Space Access (Software Wait Only) to Figure 9.10 CSn Assert Period Expansion	319 to 321	Amended 
9.5.5 MPX-I/O Interface	322	Added The data cycle is the same as that in a normal space access. The delay cycles the number of which is specified by SW[1:0] are inserted between cycle Ta3 and cycle T1. The delay cycles the number of which is specified by HW[1:0] are added after cycle T2.
Figure 9.14 Access Timing for MPX Space (Address Cycle No Wait, Assertion Extension Cycle 1.5, Data Cycle No Wait, Negation Extension Cycle 1.5)	326	Figure added
Table 9.14 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (4)-1	336	Amended

Setting

BSZ [1:0]	A2/3 ROW [1:0]	A2/3 COL [1:0]		
10 (16 Bits)	**00 (11 Bits)**	**00 (8 Bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*~~2~~	A21*~~2~~	~~A12 (BA1)~~	~~Specifies bank~~
A12	A20*2	A20*2	A11 (BA0)	Specifies bank

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>Table 9.17 Relationship between Access Size and Number of Bursts</td><td>342</td><td>Added
<table>
<tr><th>Bus Width</th><th>Access Size</th><th>Number of Bursts</th></tr>
<tr><td>16 bits</td><td>8 bits</td><td>1</td></tr>
<tr><td></td><td>16 bits</td><td>1</td></tr>
<tr><td></td><td>32 bits</td><td>2</td></tr>
<tr><td></td><td>16 bytes</td><td>8</td></tr>
<tr><td>32 bits</td><td>8 bits</td><td>1</td></tr>
<tr><td></td><td>16 bits</td><td>1</td></tr>
<tr><td></td><td>32 bits</td><td>1</td></tr>
<tr><td></td><td>16 bytes</td><td>4</td></tr>
</table></td></tr>
<tr><td>Figure 9.18 Burst Read Basic Timing (CAS Latency 1, Auto-Precharge)
to
Figure 9.25 Burst Read Timing (Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1),
Figure 9.27 Single Write Timing (Bank Active, Same Row Addresses in the Same Bank)
to
Figure 9.36 Burst ROM Access Timing (Clock Asynchronous) (Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)</td><td>343
to
352
354
to
371</td><td>Amended
D31 to D0</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
Figure 9.18 Burst Read Basic Timing (CAS Latency 1, Auto-Precharge) to Figure 9.35 Deep Power-Down Mode Transition Timing	343 to 368	Added 
Table 9.18 Access Address in SDRAM Mode Register Write • Setting for Area 2	363	Added Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460
32 bits	2	H'FFFC4880	H'0000880
	3	H'FFFC48C0	H'00008C0

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060
32 bits	2	H'FFFC4080	H'0000080
	3	H'FFFC40C0	H'00000C0

Item	Page	Revision (See Manual for Details)
• Setting for Area 3	364	Added

Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460
32 bits	2	H'FFFC5880	H'0000880
	3	H'FFFC58C0	H'00008C0

Burst read/burst write (burst length 1):

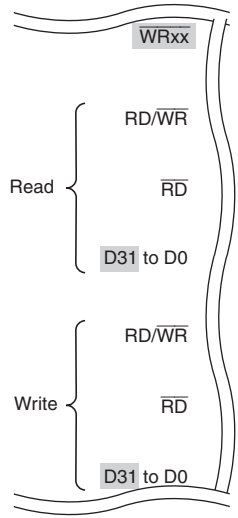
Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060
32 bits	2	H'FFFC5080	H'0000080
	3	H'FFFC50C0	H'00000C0

When a mode register write command is issued, the outputs of the external address pins are as follows.

When the data bus width of the area connected to SDRAM is 32 bits	A15 to A9	00000000 (burst read/burst write) 00000100 (burst read/single write)
	A8 to A6	010 (CAS latency 2), 011 (CAS latency 3)
	A5	0 (lap time = sequential)
	A4 to A2	000 (burst length 1)
When the data bus width of the area connected to SDRAM is 16 bits	A14 to A8	00000000 (burst read/burst write) 00000100 (burst read/single write)
	A7 to A5	010 (CAS latency 2), 011 (CAS latency 3)
	A4	0 (lap time = sequential)
	A3 to A1	000 (burst length 1)

Table 9.20 Relationship between Bus Width, Access Size, and Number of Bursts	370	Added	

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
32 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	1	1
	16 bytes※2	Not affected	4	1

Item	Page	Revision (See Manual for Details)
Figure 9.37 Basic Access Timing for SRAM with Byte Selection (BAS = 0) to Figure 9.39 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)	372 to 374	Amended 
Figure 9.40 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection	375	Figure replaced
Figure 9.41 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection	376	Figure added
Table 9.21 Conditions for Determining Number of Idle Cycles	380	Amended and added **No.** / **Condition** / **Description** (5) Read data transfer cycle — One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not B'00. Note: * This is the case for consecutive read operations when the data read are stored in separate registers.

Item	Page	Revision (See Manual for Details)
Table 9.22 Minimum Number of Idle Cycles on Internal Bus (CPU Operation), Table 9.23 Minimum Number of Idle Cycles on Internal Bus (DMAC Operation)	382	Tables replaced
Figure 9.44 Comparison between Estimated Idle Cycles and Actual Value	384	Amended

Condition	R → R	R → W	W → W	W → R
[6]	0	1	0	0
[7]	0	1	0	0
[5] + [6] + [7]	1	3	0	0
[8]	0	0	0	0
Estimated idle cycles	1	3	0	0
Actual idle cycles	1	3	0	1

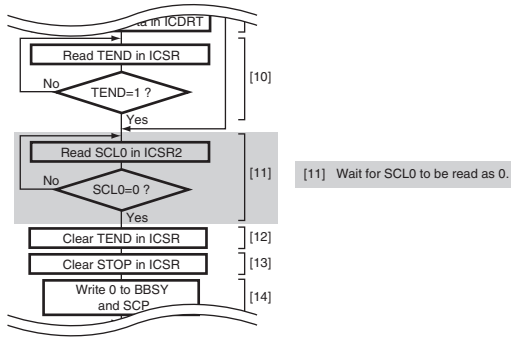
9.5.11 Bus Arbitration	385	Added and amended In bus arbitration by this LSI, it normally holds bus mastership but can release this after receiving a bus request from another device. Bus arbitration by this LSI also supports four on-chip bus masters: the CPU, DMAC, DTC, and EDMAC. The priority order of these bus masters is as follows. Bus mastership request from an external device (BREQ) > EDMAC > DTC > DMAC > CPU.
Figure 9.45 Bus Arbitration Timing	387	Amended D31 to D0

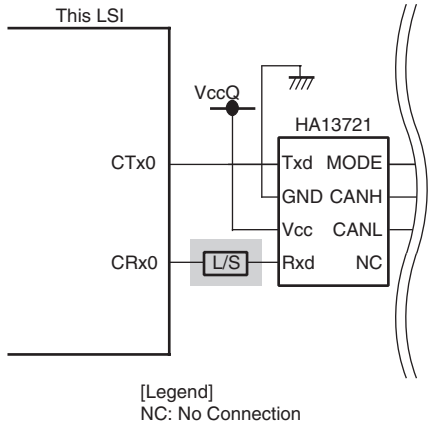
<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>Table 9.26 Number of Cycles for Access to On-Chip Memory and External Device
Figure 9.48 Timing of Write Access to Data Beyond External Bus Width When Iϕ:Bϕ = 2:1
Figure 9.49 Timing of Read Access to Data within External Bus Width When Iϕ:Bϕ = 4:1</td><td>392</td><td>Subsection, table and figure added</td></tr>
<tr><td>10.3.4 DMA Channel Control Registers (CHCR)</td><td>408</td><td>Amended
The DO, AM, AL, DL, and DS bits which specify the DREQ and DACK external pin functions can be read and written to in channels 0 to 3, but they are reserved in channels 4 to 7. The TL bit which specifies the TEND external pin function can be read and written to in channels 0 and 1, but it is reserved in channels 2 to 7. Before modifying the CHCR setting, clear the DE bit for the corresponding channel.</td></tr>
<tr><td></td><td>415</td><td>Added
<table><tr><th>Bit</th><th>Descriptions</th></tr><tr><td>0</td><td>DMA Enable
…. Clearing the DE bit to 0 can terminate the DMA transfer. Before modifying the CHCR setting, clear the DE bit to 0 for the corresponding channel.
0: DMA transfer disabled
1: DMA transfer enabled</td></tr></table></td></tr>
<tr><td>10.3.8 DMA Operation Register (DMAOR)</td><td>419</td><td>Deleted
<table><tr><th>Bit</th><th>Bit Name</th><th>Description</th></tr><tr><td>1</td><td>NMIF</td><td>[Clearing condition]
• Writing 0 after having read this bit as 1. ~~Write 1 after having read this bit as 0.~~</td></tr></table></td></tr>
<tr><td>11.3.20 Timer Output Control Register 1 (TOCR1)</td><td>521</td><td>Note 3 added</td></tr>
<tr><td>11.3.21 Timer Output Control Register 2 (TOCR2)</td><td>524</td><td>Note 2 added</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
11.3.26 Timer Cycle Data Register (TCDR)	530	Added TCDR is a 16-bit register used only in complementary PWM mode. Set half the PWM carrier sync value (note that this value should be at least double the value specified in TDDR + 3) as the TCDR register value. This register is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (decrement to increment).
11.4.4 Cascaded Operation	552	Added For simultaneous input capture of TCNT_1 and TCNT_2 during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). Edge detection as the condition for input capture is the detection of edges in the signal produced by taking the logical OR of the signals on the main and additional pins. For details, refer to (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 11.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.
Figure 11.23 Cascaded Operation Example (c)	555	Note added
11.4.5 PWM Modes • PWM mode 2	557	Amended PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by the cycle register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.
Figure 11.38 Example of Complementary PWM Mode Setting Procedure	575	Amended [8] Set the dead time in the dead time register (TDDR), 1/2 the carrier cycle in the timer cycle data register (TCDR) and timer cycle buffer register (TCBR), and 1/2 the carrier cycle plus the dead time in TGRA_3 and TGRC_3. When no dead time generation is selected, set 1 in TDDR and 1/2 the carrier cycle + 1 in TGRA_3 and TGRC_3.

Item	Page	Revision (See Manual for Details)
11.4.8 Complementary PWM Mode (2) Outline of Complementary PWM Mode Operation (j) Complementary PWM Mode PWM Output Generation Method	576	Amended A PWM waveform is generated by output of the output level selected in the timer output control register in the event of a compare-match between a counter and compare register. While TCNTS is counting, compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%.
	588	Amended If compare-match **c** occurs first following compare-match **a**, as shown in figure 11.47, compare-match **b** is ignored, and the negative phase is turned on by compare-match **d**. This is because turning off of 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 since the positive phase goes from off to off).
(2) Outline of Complementary PWM Mode Operation (g) PWM Cycle Setting	583	Added With dead time: TGRA_3 set value = TCDR set value + TDDR set value TCDR set value > Double the TDDR set value + 2 Without dead time: TGRA_3 set value = TCDR set value + 1
(k) Complementary PWM Mode 0% and 100% Duty Output	590	Amended 100% duty output is performed when the compare register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. 0% duty output is performed when the compare register value is set to the same value as TGRA_3. The waveform in this case has a positive phase with a 100% off-state.
Figure 11.110 TGI Interrupt Timing (Compare Match) (Channel 5)	637	Note added
15.6.3 Interval Timer Overflow Flag	761	Subsection added

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>Table 16.10 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)</td><td>792</td><td>Tables replaced</td></tr>
<tr><td>Table 16.11 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Clock Synchronous Mode)</td><td>793</td><td>Table added</td></tr>
<tr><td>17.3.7 Serial Status Register (SCFSR)</td><td>843</td><td>Added
<table><tr><th>Bit</th><th>Description</th></tr><tr><td>6</td><td>Note: * Do not use this bit as a transmit end flag when the DMAC/DTC writes data to SCFTDR due to a TXI interrupt request.</td></tr></table></td></tr>
<tr><td>Table 17.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (1)
to
Table 17.13 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode)</td><td>853 to 862</td><td>Tables replaced and added</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
Figure 19.19 Sample Flowchart for Master Transmit Mode	1019	Added 
19.8.2 Note on Master Receive Mode	1027	Added and amended In addition, when RCVD is set to 1 around the falling edge of the 8th clock and the receive buffer is full, a stop condition may not be issued. Use either of the following measures 1 or 2 against the situations above. 1. In master receive mode, read ICDRR before the rising edge of the 8th clock. 2. In master receive mode, set RCVD to 1 so that data is received in byte units.
20.7.6 Notes on Register Setting	1059	Subsecton added
21.3.3 RCAN-ET Control Registers (3) Bit Configuration Register (BCR0, BCR1) • Requirements of Bit Configuration Register	1087	Amended BRP: BRP[7:0] (bits 7 to 0 in BCR0)
21.8 CAN Bus Interface	1123	Added A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. The specification for this LSI circuit is a 3-V power-supply voltage, so use a level-shifter IC between its CRx0 pin and the Rxd pin of the HA13721. Figure 21.16 shows a sample connection diagram.

Item	Page	Revision (See Manual for Details)
Figure 21.16 High-Speed CAN Interface Using HA13721	1123	Added  [Legend] NC: No Connection
Table 22.7 List of pin functions in each operating mode	1135 to 1142	Table added

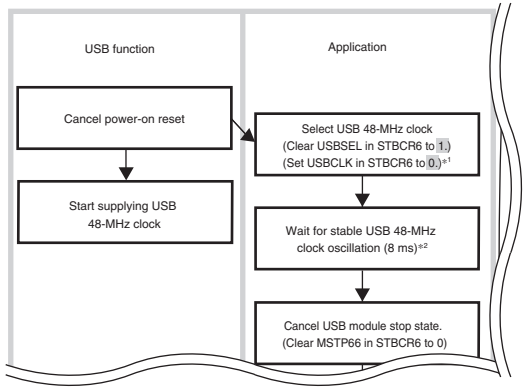
Item	Page	Revision (See Manual for Details)
22.1.5 Port B Control Registers L1 to L4 (PBCRL1 to PBCRL4) • Port B Control Register L4 (PBCRL4)	1162	Amended

Bit	Description
6 to 4	PB13 Mode Select the function of the PB13/IRQ3/$\overline{\text{POE2}}$/SDA pin. 000: PB13 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ3 input (INTC) 100: Setting prohibited 101: $\overline{\text{POE2}}$ input (POE2) 110: SDA I/O (IIC3) 111: Setting prohibited
2 to 0	PB12 Mode Select the function of the PB12/IRQ2/$\overline{\text{POE1}}$/SCL pin. 000: PB12 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ2 input (INTC) 100: Setting prohibited 101: $\overline{\text{POE1}}$ input (POE2) 110: SCL I/O (IIC3) 111: Setting prohibited

Item	Page	Revision (See Manual for Details)
22.1.6 Port B Pull-Up MOS Control Register L (PBPCRL)	1169	Amended

Bit	Description
15	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
14	
13	Reserved
12	The corresponding input pull-up MOS turns on regardless of the setting value.
11	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
10	

Item	Page	Revision (See Manual for Details)
Figure 24.2 Initial Setting	1319	Amended



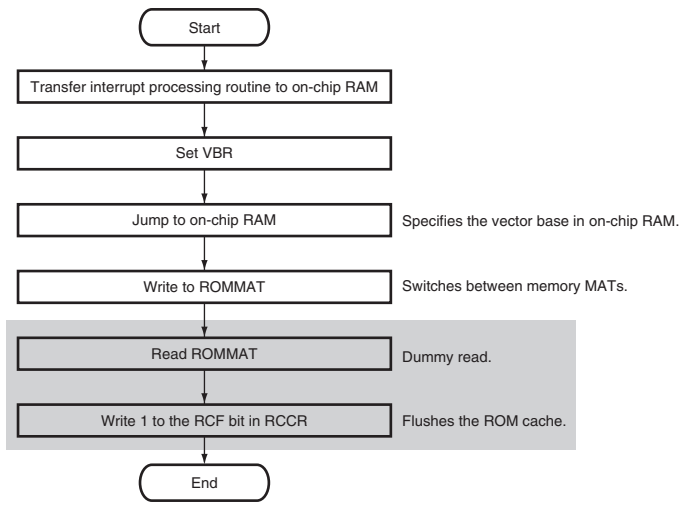
Notes: 2 The initial values of the USBSEL and USBCLK bits in STBCR6 immediately after a power-on reset are 1 and 0, respectively. Wait for the power-on oscillation settling time indicated in section 33.3.1, Clock Timing, before release from the power-on reset state. This secures the oscillation settling time for the 48-MHz USB clock. After halting the clock to change the values of the USBSEL and USBCLK bits, secure the oscillation settling time when restarting the clock.

Figure 24.17 Example of DMA Transfer (Channel 0) for Bulk-OUT Transfer (EP1) (When Receive Data Size is Determined Before Receiving OUT Token) Figure 24.20 Example of DMA Transfer (Channel 0) for Bulk-IN Transfer (EP2) (When Transmit Data Size is Determined Before Receiving IN Token)	1340, 1344	Figure replaced

Item	Page	Revision (See Manual for Details)
26.3.1 Descriptor Lists and Data Buffers (1) Transmit Descriptor	1449	Amended 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 below 42 bytes, operation cannot be guaranteed.
27.5.4 USB Boot Mode (3) Notes when Executing USB Boot Mode	1503	Amended • To maintain stable power supply when programming or erasing flash memory, the cable should not be connected via the bus-powered hub.
Table 27.14 Error Protection Types	1569	Amended

Error	Description
Illegal command error	An undefined code has been specified in the first cycle of an FCU command.
	The value specified in the last of the multiple cycles of an FCU command is not H'D0.
	The peripheral clock specified in PCKAR is not in the range from 1 to 100 MHz.
	The command issued during programming or erasure is not a suspend command.

Figure 27.36 Example of MAT Switching Steps	1572	Amended



Item	Page	Revision (See Manual for Details)
27.10.10 Items Prohibited during Programming and Erasure	1575	Added • Cutting off the power supply • Transitions to software standby mode • Read access to the flash memory by the CPU, DMAC or DTC • Writing a new value to the FRQCR register • Setting the PCKAR register for a different frequency from that of Pϕ.
27.10.11 Abnormal Ending of Programming or Erasure	1575	Subsection added
28.1 Features • Blank check function	1580	Added Blank checking proceeds for areas where erasure has been completed normally to confirm that the data have actually been erased. When erasure or programming in progress is stopped (e.g. by input of the reset signal or shutting down the power), blank checking cannot be used to check whether the data have actually been erased or written.
28.8.9 Items Prohibited during Programming and Erasure	1619	Added • Cutting off the power supply • Transitions to software standby mode • Read access to the flash memory by the CPU, DMAC or DTC • Writing a new value to the FRQCR register • Setting the PCKAR register for a different frequency from that of Pϕ.
28.8.10 Abnormal Ending of Programming or Erasure 28.8.11 Handling when Erasure or Programming is Stopped	1619	Subsection added
Table 30.4 Register States in Software Standby Mode	1641	Amended

Module Name	Initialized Registers	Registers Whose Content is Retained
Compare match timer (CMT)	—	All registers

Item	Page	Revision (See Manual for Details)
Table 33.6 Control Signal Timing	1776	Added

Item	Symbol	Bφ 50MHz		Unit	Figure
		Min.	Max.		
$\overline{RES}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*2*4		tcyc	Figures 33.3 to 33.6
		1.5*4		µs	
$\overline{RES}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100		µs	

Note: 4 Input the reset pulse over t_{RESW1} so that all conditions are met.

Figure 33.13 Basic Bus Timing for Normal Space (One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)	1786	Amended

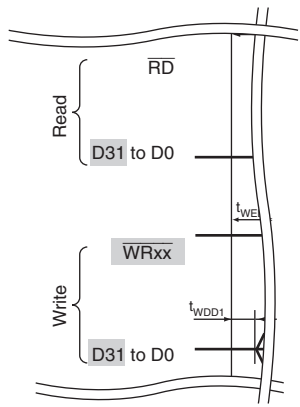


Table 33.6 Control Signal Timing	1776	Added

Item	Symbol	Bϕ = 50 MHz		Unit	Figure
		Min.	Max.		
$\overline{RES}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*4	—	t_{cyc}	Figures 33.3 to 33.6
		1.5*4	—	µs	
$\overline{RES}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100	—	µs	

Notes: 4. Input the reset pulse over t_{RESW1} so that all conditions are met.

Item	Page	Revision (See Manual for Details)
Figure 33.50 SPI Timing (Master, CPHA = 0)	1823	Added and amended

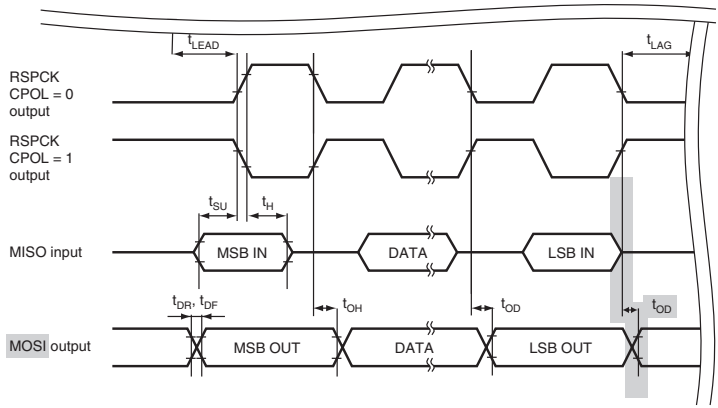


Figure 33.51 SPI Timing (Master, CPHA = 1)	1823	Amended

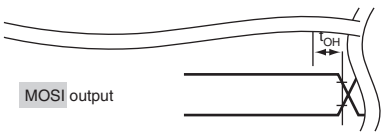


Figure 33.52 SPI Timing (Slave, CPHA = 0)	1824	Amended

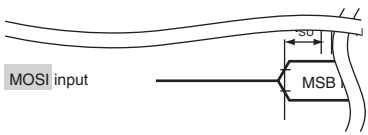


Figure 33.53 SPI Timing (Slave, CPHA = 1)	1824	Amended

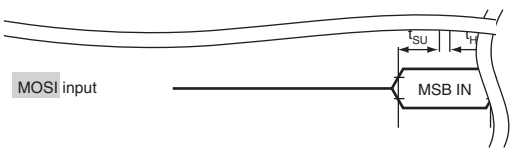
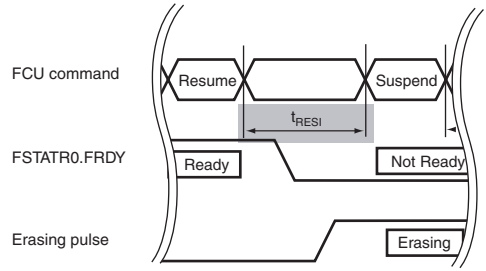


Table 33.25 ROM (Flash Memory for Code Storage) Characteristics, Table 33.26 FLD (Flash Memory for Data Storage) Characteristics	1840	Added

Item	Symbol	Min.	Typ.	Max.	Unit
Resume command interval time	t_{RESI}	1.7	—	—	ms

Item	Page	Revision (See Manual for Details)
Figure 33.71 Flash Programming/Erasing Suspend Timing	1841	Added • Erasing suspend in suspend priority mode 
Appendix Table A.1 Pin States	1845 to 1853	Amended

Pin State			
Reset State			
Power-On			
Expansion without ROM		Expansion with ROM	Single Chip
16 Bits	32 Bits		

Index

1

16-bit/32-bit displacement 38

A

A/D conversion time 1051
A/D converter (ADC) 1031
A/D converter activation 633
A/D converter activation by MTU2 and MTU2S 1052
A/D converter characteristics 1837
A/D converter start request delaying function 614
A/D trigger input timing 1828
Absolute accuracy 1056
Absolute address 38
Absolute address accessing 38
Absolute maximum ratings 1767
AC characteristics 1772
AC characteristics measurement conditions 1836
Access size and data alignment 310
Access wait control 319
Accessing MII Registers 1403
Address errors 115
Address map 257
Address multiplexing 330
Addressing modes 39
Arithmetic operation instructions 59
Auto-refreshing 356
Auto-request mode 427

B

Banked register and input/output of banks 166
Bit manipulation instructions 70
Bit synchronous circuit 1025
Block transfer mode 233
Boot mode 1496, 1598
Branch instructions 64
Break detection and processing 828, 893
Break on data access cycle 201
Break on instruction fetch cycle 200
Burst mode 440
Burst ROM (clock asynchronous) interface 369
Burst ROM (clock synchronous) interface 376
Bus arbitration 385
Bus connections in RAM 1622
Bus state controller (BSC) 253
Bus timing 1780

C

Calculating exception handling vector table addresses 110
CAN interface 1066
CAN sleep mode 1109
Canceling Software Standby Mode 757
Caution on period setting 649
Chain transfer 234
Changing the Frequency 96
Clock frequency control circuit 83
Clock operating modes 86
Clock pulse generator (CPG) 81
Clock timing 1773
Clocked synchronous serial format 1014

CMCNT count timing 743
Compare match timer (CMT) 737
Complementary PWM mode 572
Conflict between byte-write and count-up processes of CMCNT 748
Conflict between NMI Interrupt and DTC Activation 251
Conflict between word-write and count-up processes of CMCNT 747
Conflict between write and compare-match processes of CMCNT 746
Connection to the PHY-LSI 1409
Continuous scan mode 1047
Control signal timing 1776
Controller area network (RCAN-ET) 1063
Controlling RSPI pins 933
CPU 23
Crystal oscillator 83
CSn assert period expansion 321
Cycle steal mode 438

D

Data format 23
Data format in registers 33
Data formats in memory 33
Data transfer controller (DTC) 207
Data transfer instructions 55
Data transfer with interrupt request signals 170
DC characteristics 1768
Dead time compensation 626
Definition of time quanta 1084
Definitions of A/D conversion accuracy 1056
Delayed branch instructions 36
Direct memory access controller (DMAC) 397
Displacement accessing 39
Divider 83
DMA transfer flowchart 426
DMAC and DTC activation 632
DMAC module timing 1811
DREQ pin sampling timing 443
DTC activation by interrupt 246
DTC activation sources 219
DTC execution status 239
DTC vector address 221
Dual address mode 435

E

Effective address calculation 39
Electrical characteristics 1767
Endian 310
Equation for getting SCBRR value 851
Error protection 1568, 1615
Error protection types 1568
EtherC receiver 1399
EtherC transmitter 1396
Ethernet Controller (EtherC) 1359
Ethernet Controller Direct Memory Access Controller (E-DMAC) 1411
Example of Relationship between Clock Operating Mode and Frequency Range 87
Example of USB External Circuitry 1353
Exception handling 105
Exception handling state 73
Exception handling vector table 109
Exception source generation immediately after delayed branch instruction 125
Exceptions triggered by instructions 121
External pulse width measurement 625
External request mode 427
External trigger input timing 1052

F

FCU command list 1534
FCU command usage 1541, 1609
Fixed mode 431
FLD 1577
Floating-point operation instructions 68
Floating-point registers 28
Floating-point system registers 29
Flow Control 1408
FPU-related CPU instructions 70
Full-scale error 1056

G

General illegal instructions 123
General registers 24
Global base register (GBR) 27

H

Halt mode 1109
Hardware protection 1566, 1614
H-UDI interrupt 143
H-UDI related pin timing 1834

I

I/O port timing 1829
I/O ports 1225
I2C bus format 1004
I2C bus interface 3 (IIC3) 985
ID Reorder 1076
IIC3 module timing 1826
Immediate data 37
Immediate data accessing 37
Immediate data format 34
Initial values of control registers 32
Initial values of floating-point registers 32
Initial values of floating-point system registers 32
Initial values of general registers 32
Initial values of system registers 32
Initializing RSPI 960
Input sampling and A/D conversion time 1050
Instruction features 35
Instruction format 44
Instruction set 48
Integer division instructions 123
Interrupt controller (INTC) 129
Interrupt exception handling 120
Interrupt exception handling vectors and priorities 147
Interrupt priority level 119
Interrupt response time 159
IRQ interrupts 144

J

Jump table base register (TBR) 27

L

Load-store architecture 35
Local acceptance filter mask (LAFM) 1074
Location of transfer information and DTC vector table 219
Logic operation instructions 62

M

Magic Packet Detection 1406
Mailbox 1066
Mailbox control 1066
Mailbox structure 1069
Manual reset 113, 1630
Master receive operation 1007
Master transmit operation 1005
MCU extension mode 76
MCU operating modes 75
Message control field 1070
Message data fields 1075
Message receive sequence 1116
Message transmission sequence 1113
Micro processor interface (MPI) 1066
MII Frame Timing 1401
Module standby function 1644
Module standby mode setting 249, 830
MOSI signal value determination during SSL negate period 934
MPX-I/O interface 322
MTU2 functions 452
MTU2 interrupts 631
MTU2 output pin initialization 666
MTU2 timing 1812
MTU2–MTU2S synchronous operation 619
MTU2S functions 699
MTU2S timing 1814
Multi-function timer pulse unit 2 (MTU2) 451
Multi-function timer pulse unit 2S (MTU2S) 699
Multiply and accumulate register high (MACH) 27
Multiply and accumulate register low (MACL) 27
Multiply/multiply-and-accumulate operations 36
Multiprocessor communication function 818

N

NMI interrupt 143
Noise filter 1017
Nonlinearity error 1056
Normal space interface 314
Normal transfer mode 230
Note on changing operating mode 80
Note on using an external crystal resonator 103
Notes on board design 1058
Notes on Connecting Capacitors 1844
Notes on noise countermeasures 1059
Notes on register setting 1059

O

Offset error 1056
On-chip peripheral module interrupts 145
On-chip peripheral module request 429
On-chip RAM address space 1623
Operation by IPG Setting 1407
Operation in asynchronous mode 872
Operation in clocked synchronous mode 882
Output load circuit 1836

P

Page conflict 1628
Pin function controller (PFC) 1127
PLL circuit 83
POE2 interrupt source 735
POE2 module timing 1815
Port output enable 2 (POE2) 707
Power-down modes 1629
Power-down state 73
Power-on reset 1630
Procedure register (PR) 28
Program counter (PC) 28
Program execution state 73

Programmer Mode 1566
Programming/erasing host command wait state 1522
Protection 1566, 1614

Q

Quantization error 1056

R

RAM 1621
RAM block diagram 1622
RCAN-ET bit rate calculation 1087
RCAN-ET interrupt sources 1120
RCAN-ET memory map 1067
RCAN-ET reset sequence 1108
Receive data sampling timing and receive margin (asynchronous mode) 894
Receive Descriptor 0 (RD0) 1455
Receive Descriptor 1 (RD1) 1457
Receive Descriptor 2 (RD2) 1457
Reconfiguration of Mailbox 1118
Register addresses (by functional module, in order of the corresponding section numbers) 1676
Register bank error exception handling 117, 169
Register bank errors 117
Register bank exception 169
Register banks 32, 165
Register bits 1704
Register states in each operating mode 1744
Register states in software standby mode 1641
Registers
ABACK0 1101
ACLKCR 94
ADANSR_0 to ADANSR_1 1041
ADBYPSCR_0 to ADBYPSCR_1 1042
ADCR_0 to ADCR_1 1035
ADDR0 to ADDR7 1043
ADSR_0 to ADSR_1 1038
ADSTRGR_0 to ADSTRGR_1 1039
APR 1389
BAMR 180, 184, 188, 192
BAR 179, 183, 187, 191
BBR 181, 185, 189, 193
BCFRR 1395
BCR0, BCR1 1084
BRCR 195
BSBPR 1654
BSBSR 1655
BSCEHR 219, 307
BSID 1654
BSIR 1654
CDCR 1379
CEFCR 1382
CHCR 408
CMCNT 742
CMCOR 742
CMCSR 740
CMNCR 262
CMSTR 739
CNDCR 1381
CRA 214
CRB 215
CSBCR 265
CSWCR 270
DAR (DMAC) 406
DAR (DTC) 213
DMAOR 419
DMARS0 to DMARS3 423
DMATCR 407
DTCCR 217
DTCERA to DTCERE 216

DTCVBR 218
ECMR 1365
ECSIPR 1371
ECSR 1369
EDMR 1414
EDOCR 1447
EDRRR 1416
EDTRR 1415
EEPBCCNT 1594
EEPBCSTAT 1595
EEPRE0 1590
EEPWE0 1591
EESIPR 1424
EESR 1419
FAEINT 1476, 1588
FASTAT 1474, 1585
FCFTR 1443
FCMDR 1488
FCPSR 1489
FCURAME 1478
FDR 1433
FENTRYR 1484, 1592
FMODR 1473, 1584
FPESTAT 1490
FPMON 1472
FPROTR 1486
FPSCR 30
FPUL 29
FRECR 1383
FRESETR 1487
FRQCR 90
FSTATR0 1479
FSTATR1 1483
GSR 1081
HCPCR 1211
IBCR 139
IBNR 140
ICCR1 989
ICCR2 992
ICDRR 1002
ICDRS 1002
ICDRT 1001
ICIER 996
ICMR 994
ICR0 135
ICR1 136
ICSR 998
ICSR1 712
ICSR2 717
ICSR3 720
IFCR 1213
IMR 1094
IOSR 1446
IPGR 1388
IPR01, IPR02, IPR05 to IPR19 133
IRQRR 137
IRR 1089
LCCR 1380
MAFCR 1387
MAHR 1374
MALR 1375
MBIMR0 1104
MCLKCR 93
MCR 1076
MPR 1390
MRA 210
MRB 211
NF2CYC 1003
OCSR1 716
OCSR2 718
OSCCR 95
PACRH1 1147
PACRH2 1146
PACRL1 1156
PACRL2 1154
PACRL3 1152
PACRL4 1150
PADRH 1226
PADRL 1226
PAIORH 1145

PAIORL 1145
PAPCRH 1158
PAPCRL 1159
PAPRH 1228
PAPRL 1228
PBCRL1 1167
PBCRL2 1165
PBCRL3 1163
PBCRL4 1160
PBDRL 1231
PBIORL 1160
PBPCRL 1169
PBPRL 1232
PCCRL1 1177
PCCRL2 1175
PCCRL3 1173
PCCRL4 1170
PCDRL 1234
PCIORL 1170
PCKAR 1492
PCPCRL 1179
PCPRL 1236
PDACKCR 1214
PDCRH1 1188
PDCRH2 1186
PDCRH3 1183
PDCRH4 1181
PDCRL1 1196
PDCRL2 1194
PDCRL3 1192
PDCRL4 1190
PDDRH 1238
PDDRL 1240
PDIORH 1180
PDIORL 1180
PDPCRH 1198
PDPCRL 1199
PDPRH 1241
PDPRL 1242
PECRL1 1207
PECRL2 1205
PECRL3 1203
PECRL4 1201
PEDRL 1244
PEIORL 1200
PEPCRL 1210
PEPRL 1245
PFDRL 1247
PIR 1373
POECR1 723
POECR2 725
PSR 1377
RBWAR 1439
RCCR 1491
RDAR 417
RDFAR 1440
RDLAR 1418
RDMATCR 418
RDMLR 1392
REC 1094
RFCF 1393
RFCR 1386
RFLR 1376
RFOCR 1438
RFPR0 1103
RMCR 1435
RMFCR 1430
ROMMAT 1477
RSAR 416
RTCNT 305
RTCOR 306
RTCSR 303
RXPR0 1102
SAR (DMAC) 405
SAR (DTC) 212
SAR (IIC3) 1001
SCBRR (SCI) 784
SCBRR (SCIF) 851
SCFCR 863
SCFDR 865

SCFRDR ... 834
SCFSR ... 843
SCFTDR ... 835
SCLSR ... 868
SCRDR (SCI) ... 767
SCRSR (SCI) ... 767
SCRSR (SCIF) ... 834
SCSCR (SCI) ... 772
SCSCR (SCIF) ... 839
SCSDCR ... 783
SCSEMR ... 869
SCSMR (SCI) ... 768
SCSMR (SCIF) ... 836
SCSPTR (SCI) ... 781
SCSPTR (SCIF) ... 866
SCSSR ... 775
SCTDR (SCI) ... 768
SCTSR (SCI) ... 768
SCTSR (SCIF) ... 835
SDCR ... 299
SDID ... 1666
SDIR ... 1666
SPBR ... 918
SPCKD ... 923
SPCMD ... 926
SPCR ... 903
SPDCR ... 919
SPDR ... 913
SPND ... 925
SPOER ... 722
SPPCR ... 907
SPSCR ... 915
SPSR ... 908
SPSSR ... 916
SSLND ... 924
SSLP ... 906
STBCR ... 1632
STBCR2 ... 1633
STBCR3 ... 1634
STBCR4 ... 1636
STBCR5 ... 1637
STBCR6 ... 1638
SYSCR1 ... 1624
SYSCR2 ... 1626
TADCOBRA_4 ... 509
TADCOBRB_4 ... 509
TADCORA_4 ... 509
TADCORB_4 ... 509
TADCR ... 506
TBRAR ... 1441
TBTER ... 534
TBTM ... 501
TCBR ... 531
TCDR ... 530
TCNT ... 510
TCNTCMPCLR ... 488
TCNTS ... 529
TCR ... 462
TCSYSTR ... 515
TDDR ... 530
TDER ... 536
TDFAR ... 1442
TDLAR ... 1417
TEC ... 1094
TFTR ... 1431
TFUCR ... 1437
TGCR ... 527
TGR ... 510
TICCR ... 503
TIER ... 489
TIOR ... 469
TITCNT ... 533
TITCR ... 531
TLFRCR ... 1385
TMDR ... 466
TOCR1 ... 520
TOCR2 ... 523
TOER ... 519
TOLBR ... 526
TPAUSECR ... 1394

TPAUSER 1391
TRIMD 1445
TROCR 1378
TRSCER 1427
TRWER 518
TSFRCR 1384
TSR 494
TSTR 511
TSYCRS 504
TSYR 513
TWCR 537
TXACK0 1100
TXCR0 1099
TXPR1, TXPR0 1097
UMSR0 1105
USBCTLR 1306
USBCVR 1305
USBDASTS0 1279
USBDASTS1 1280
USBDASTS2 1281
USBDASTS3 1282
USBDMAR 1302
USBEPDR0i 1271
USBEPDR0o 1271
USBEPDR0s 1272
USBEPDR1 1273
USBEPDR2 1273
USBEPDR3 1274
USBEPDR4 1274
USBEPDR5 1275
USBEPDR6 1275
USBEPDR7 1276
USBEPDR8 1276
USBEPDR9 1277
USBEPIR 1307
USBEPSTL0 1291
USBEPSTL1 1292
USBEPSTL2 1293
USBEPSTL3 1294
USBEPSZ0o 1277
USBEPSZ1 1278
USBEPSZ4 1278
USBEPSZ7 1279
USBFCLR0 1287
USBFCLR1 1288
USBFCLR2 1289
USBFCLR3 1290
USBIER0 1261
USBIER1 1262
USBIER2 1263
USBIER3 1264
USBIER4 1265
USBIFR0 1254
USBIFR1 1255
USBIFR2 1257
USBIFR3 1258
USBIFR4 1260
USBISR0 1266
USBISR1 1267
USBISR2 1268
USBISR3 1269
USBISR4 1270
USBSTLSR1 1296
USBSTLSR2 1298
USBSTLSR3 1300
USBTRG0 1283
USBTRG1 1284
USBTRG2 1285
USBTRG3 1286
USBTRNTREG0 1312
USBTRNTREG1 1314
USDTENDRR 141
WRCSR 754
WTCNT 751
WTCSR 752
Relationship between RSPI modes and SPCR and description of each mode 931
Renesas serial peripheral interface (RSPI) 897
Repeat transfer mode 231

Reset state 73
Reset-synchronized PWM mode 569
Restoration from bank 167
Restoration from stack 168
Restriction on DMAC and DTC usage ... 893
RISC-type instruction set 35
ROM 1465
Round-robin mode 431
RSPI data format 947
RSPI error detection function 955
RSPI system configuration example 935
RSPI Timing 1821
RSPI transfer format 945, 946

S

Saving to bank 166
Saving to stack 168
SCI interrupt sources 824
SCIF interrupt sources 891
SCIF module timing 1819
SCSPTR and SCI pins 825
SDRAM interface 327
Self-refreshing 358
Sending a break signal 828, 893
Serial communication interface (SCI) 763
Serial communication interface with FIFO (SCIF) 831
Shift instructions 63
Sign extension of word data 35
Single address mode 437
Single chip mode 76
Single-cycle scan mode 1044
Slave receive operation 1012
Slave transmit operation 1009
Sleep mode 1640
Slot illegal instructions 122
Software protection 1567, 1614
Software standby mode 1640
SRAM interface with byte selection 371
Stack status after exception handling ends 126
Standby control circuit 83
State transition in boot mode 1497
Status register (SR) 26
Supported DMA transfers 434
Suspending operation 1560
System control instructions 66

T

T bit 36
Test mode settings 1106
The address map for each mailbox 1068
The address map for the operating modes 77
Timing to clear an interrupt source 173
Transfer information read skip function . 229
Transfer information writeback skip function 230
Transmission buffer empty/ receive buffer full flags 953
Transmit Descriptor 0 (TD0) 1451
Transmit Descriptor 1 (TD1) 1453
Transmit Descriptor 2 (TD2) 1453
Transmit/receive processing of multi-buffer frame 1462
Trap instructions 122

U

UBC trigger timing 1810
Unconditional branch instructions with no delay slot 36
USB Function Module (USB) 1249
User boot mode 1563

User break controller (UBC) 175
User break interrupt 143
User debugging interface (H-UDI) 1647
User program mode 1534
Using interval timer mode 759
Using watchdog timer mode 757

V

Vector base register (VBR) 27

W

Wait between access cycles 377
Watchdog timer (WDT) 749
Watchdog timer timing 1816

SH7214 Group, SH7216 Group User's Manual: Hardware

Publication Date: Rev.1.01 May 22, 2009
Rev.4.00 Jun 21, 2013

Published by: Renesas Electronics Corporation



SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2013 Renesas Electronics Corporation. All rights reserved.

Colophon 1.3

SH7214 Group, SH7216 Group

Renesas Electronics Corporation

R01UH0230EJ0400