User's Manual

32

SH7239 Group, SH7237 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine family

Renesas Electronics
www.renesas.com

Rev.1.00 Sep 2010

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 SH7239 and SH7237 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	SH7239 Group, SH7237 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	REJ09B0086
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

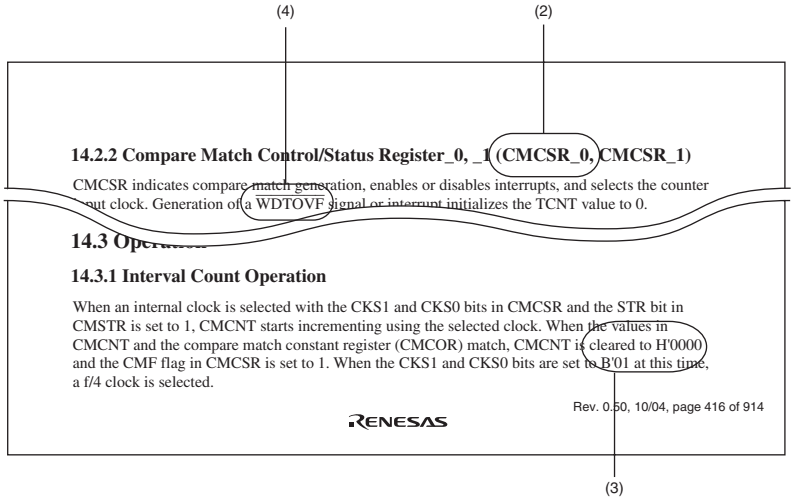
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(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] $\overline{\text{WDTOVF}}$

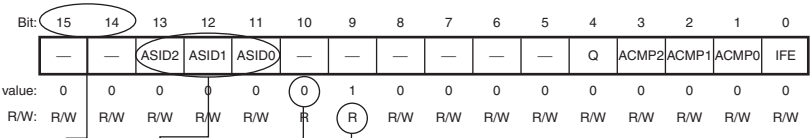


Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

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 communication interface adapter
bps	Bits per second
CRC	Cyclic redundancy check
DMA	Direct memory access
DMAC	Direct memory access controller
GSM	Global System for Mobile Communications
Hi-Z	High impedance
IEBus	Inter Equipment Bus
I/O	Input/output
IrDA	Infrared Data Association
LSB	Least significant bit
MSB	Most significant bit
NC	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

Contents

Section 1 Overview 1
1.1 Features 1
1.2 List of Products 9
1.3 Block Diagram 10
1.4 Pin Assignment 11
1.5 Pin Functions 13

Section 2 CPU 21
2.1 Data Format 21
2.2 Register Descriptions 21
2.2.1 General Registers 21
2.2.2 Control Registers 22
2.2.3 System Registers 24
2.2.4 Floating-Point Registers (SH7239 Group Only) 25
2.2.5 Floating-Point System Registers (SH7239 Group Only) 26
2.2.6 Register Bank 29
2.2.7 Initial Values of Registers 29
2.3 Data Formats 30
2.3.1 Data Format in Registers 30
2.3.2 Data Formats in Memory 30
2.3.3 Immediate Data Format 31
2.4 Instruction Features 32
2.4.1 RISC-Type Instruction Set 32
2.4.2 Addressing Modes 36
2.4.3 Instruction Format 41
2.5 Instruction Set 45
2.5.1 Instruction Set by Classification 45
2.5.2 Data Transfer Instructions 52
2.5.3 Arithmetic Operation Instructions 56
2.5.4 Logic Operation Instructions 59
2.5.5 Shift Instructions 60
2.5.6 Branch Instructions 61
2.5.7 System Control Instructions 63
2.5.8 Floating-Point Operation Instructions (SH7239 Group Only) 65
2.5.9 FPU-Related CPU Instructions (SH7239 Group Only) 67
2.5.10 Bit Manipulation Instructions 68

2.6 Processing States 69

Section 3 MCU Operating Modes 71
3.1 Selection of Operating Modes 71
3.2 Input/Output Pins 72
3.3 Operating Modes 72
3.3.1 Mode 2 (MCU Extension Mode 2) 72
3.3.2 Mode 3 (Single Chip Mode) 72
3.4 Address Map 73
3.5 Initial State in This LSI 75
3.6 Note on Changing Operating Mode 75

Section 4 Clock Pulse Generator (CPG) 77
4.1 Features 77
4.2 Input/Output Pins 81
4.3 Clock Operating Modes 82
4.4 Register Descriptions 86
4.4.1 Frequency Control Register (FRQCR) 86
4.4.2 MTU Clock Frequency Control Register (MCLKCR) 89
4.4.3 AD Clock Frequency Control Register (ACLKCR) 90
4.4.4 Oscillation Stop Detection Control Register (OSCCR) 91
4.5 Changing the Frequency 92
4.6 Oscillator 93
4.6.1 Connecting Crystal Resonator 93
4.6.2 External Clock Input Method 94
4.7 Oscillation Stop Detection 95
4.8 Notes on Board Design 96
4.8.1 Note on Using an External Crystal Resonator 96

Section 5 Exception Handling 99
5.1 Overview 99
5.1.1 Types of Exception Handling and Priority 99
5.1.2 Exception Handling Operations 101
5.1.3 Exception Handling Vector Table 103
5.2 Resets 105
5.2.1 Types of Reset 105
5.2.2 Power-On Reset 106
5.2.3 Manual Reset 107
5.3 Address Errors 109
5.3.1 Address Error Sources 109

5.3.2 Address Error Exception Handling 110
5.4 Register Bank Errors 111
5.4.1 Register Bank Error Sources 111
5.4.2 Register Bank Error Exception Handling 111
5.5 Interrupts 112
5.5.1 Interrupt Sources 112
5.5.2 Interrupt Priority Level 113
5.5.3 Interrupt Exception Handling 114
5.6 Exceptions Triggered by Instructions 115
5.6.1 Types of Exceptions Triggered by Instructions 115
5.6.2 Trap Instructions 116
5.6.3 Slot Illegal Instructions 116
5.6.4 General Illegal Instructions 117
5.6.5 Integer Division Instructions 117
5.6.6 Floating Point Operation Instruction (SH7239 Group Only) 118
5.7 When Exception Sources Are Not Accepted 119
5.8 Stack Status after Exception Handling Ends 120
5.9 Usage Notes 122
5.9.1 Value of Stack Pointer (SP) 122
5.9.2 Value of Vector Base Register (VBR) 122
5.9.3 Address Errors Caused by Stacking of Address Error Exception Handling 122

Section 6 Interrupt Controller (INTC) 123
6.1 Features 123
6.2 Input/Output Pins 125
6.3 Register Descriptions 126
6.3.1 Interrupt Priority Registers 01, 02, 05 to 18 (IPR01, IPR02, IPR05 to IPR18) 127
6.3.2 Interrupt Control Register 0 (ICR0) 129
6.3.3 Interrupt Control Register 1 (ICR1) 130
6.3.4 IRQ Interrupt Request Register (IRQRR) 131
6.3.5 Bank Control Register (IBCR) 133
6.3.6 Bank Number Register (IBNR) 134
6.4 Interrupt Sources 136
6.4.1 NMI Interrupt 136
6.4.2 User Break Interrupt 136
6.4.3 H-UDI Interrupt 136
6.4.4 IRQ Interrupts 137
6.4.5 Memory Error Interrupt 137
6.4.6 On-Chip Peripheral Module Interrupts 138

6.5 Interrupt Exception Handling Vector Table and Priority 139
6.6 Operation 147
6.6.1 Interrupt Operation Sequence 147
6.6.2 Stack after Interrupt Exception Handling 150
6.7 Interrupt Response Time 151
6.8 Register Banks 157
6.8.1 Banked Register and Input/Output of Banks 158
6.8.2 Bank Save and Restore Operations 158
6.8.3 Save and Restore Operations after Saving to All Banks 160
6.8.4 Register Bank Exception 161
6.8.5 Register Bank Error Exception Handling 161
6.9 Interrupt Requests 162
6.9.1 Handling Interrupt Request Signals as DTC Activating Sources and CPU Interrupt Sources but Not as DMAC Activating Sources 164
6.9.2 Handling Interrupt Request Signals as DMAC Activating Sources but Not as CPU Interrupt Sources 164
6.9.3 Handling Interrupt Request Signals as DTC Activating Sources but Not as CPU Interrupt Sources or DMAC Activating Sources 164
6.9.4 Handling Interrupt Request Signals as CPU Interrupt Sources but Not as DTC Activating Sources or DMAC Activating Sources 165
6.10 Usage Note 165
6.10.1 Timing to Clear an Interrupt Source 165
6.10.2 In Case the NMI Pin is not in Use 165

Section 7 User Break Controller (UBC) 167
7.1 Features 167
7.2 Input/Output Pin 169
7.3 Register Descriptions 170
7.3.1 Break Address Register_0 (BAR_0) 171
7.3.2 Break Address Mask Register_0 (BAMR_0) 172
7.3.3 Break Bus Cycle Register_0 (BBR_0) 173
7.3.4 Break Address Register_1 (BAR_1) 175
7.3.5 Break Address Mask Register_1 (BAMR_1) 176
7.3.6 Break Bus Cycle Register_1 (BBR_1) 177
7.3.7 Break Address Register_2 (BAR_2) 179
7.3.8 Break Address Mask Register_2 (BAMR_2) 180
7.3.9 Break Bus Cycle Register_2 (BBR_2) 181
7.3.10 Break Address Register_3 (BAR_3) 183
7.3.11 Break Address Mask Register_3 (BAMR_3) 184
7.3.12 Break Bus Cycle Register_3 (BBR_3) 185

7.3.13 Break Control Register (BRCR) 187
7.4 Operation 191
7.4.1 Flow of the User Break Operation 191
7.4.2 Break on Instruction Fetch Cycle 192
7.4.3 Break on Data Access Cycle 193
7.4.4 Value of Saved Program Counter 194
7.4.5 Usage Examples 195
7.5 Interrupt Source 197
7.6 Usage Notes 198

Section 8 Data Transfer Controller (DTC) 199
8.1 Features 199
8.2 Register Descriptions 201
8.2.1 DTC Mode Register A (MRA) 202
8.2.2 DTC Mode Register B (MRB) 203
8.2.3 DTC Source Address Register (SAR) 204
8.2.4 DTC Destination Address Register (DAR) 205
8.2.5 DTC Transfer Count Register A (CRA) 206
8.2.6 DTC Transfer Count Register B (CRB) 207
8.2.7 DTC Enable Registers A to E (DTCERA to DTCERE) 208
8.2.8 DTC Control Register (DTCCR) 209
8.2.9 DTC Vector Base Register (DTCVBR) 210
8.2.10 Bus Function Extending Register (BSCEHR) 211
8.3 Activation Sources 211
8.4 Location of Transfer Information and DTC Vector Table 211
8.5 Operation 216
8.5.1 Transfer Information Read Skip Function 221
8.5.2 Transfer Information Write-Back Skip Function 222
8.5.3 Normal Transfer Mode 222
8.5.4 Repeat Transfer Mode 223
8.5.5 Block Transfer Mode 225
8.5.6 Chain Transfer 226
8.5.7 Operation Timing 228
8.5.8 Number of DTC Execution Cycles 231
8.5.9 DTC Bus Release Timing 234
8.5.10 DTC Activation Priority Order 236
8.6 DTC Activation by Interrupt 238
8.7 Examples of Use of the DTC 239
8.7.1 Normal Transfer Mode 239
8.7.2 Chain Transfer when Transfer Counter = 0 240

8.8 Interrupt Sources 241
8.9 Usage Notes 241
8.9.1 Module Standby Mode Setting 241
8.9.2 On-Chip RAM 242
8.9.3 DTCE Bit Setting 242
8.9.4 Chain Transfer 242
8.9.5 Transfer Information Start Address, Source Address, and Destination Address 242
8.9.6 Access to DTC Registers through DTC 242
8.9.7 Notes on IRQ Interrupt as DTC Activation Source 242
8.9.8 Note on SCI or SCIF as DTC Activation Sources 243
8.9.9 Clearing Interrupt Source Flag 243
8.9.10 Conflict between NMI Interrupt and DTC Activation 243
8.9.11 Operation when a DTC Activation Request has been Cancelled 243
8.9.12 Note on Writing to DTCER 243

Section 9 Bus State Controller (BSC) (SH7239A and SH7237A only) 245
9.1 Features 245
9.2 Input/Output Pins 247
9.3 Area Overview 247
9.3.1 Address Map 247
9.3.2 Setting Operating Modes 248
9.4 Register Descriptions 249
9.4.1 Common Control Register (CMNCR) 250
9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 1, 3 to 6) 253
9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 1, 3 to 6) 258
9.4.4 Bus Function Extending Register (BSCEHR) 271
9.5 Operation 274
9.5.1 Endian/Access Size and Data Alignment 274
9.5.2 Normal Space Interface 277
9.5.3 Access Wait Control 281
9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion 283
9.5.5 MPX-I/O Interface 284
9.5.6 Wait between Access Cycles 288
9.5.7 Bus Arbitration 295
9.5.8 Others 297
9.6 Usage Note 301
9.6.1 Note on Connection of External LSI Circuits such as SRAMs and ASICs 301

Section 10 Direct Memory Access Controller (DMAC) 303
10.1 Features 303
10.2 Input/Output Pins 305
10.3 Register Descriptions 306
10.3.1 DMA Source Address Registers (SAR) 311
10.3.2 DMA Destination Address Registers (DAR) 312
10.3.3 DMA Transfer Count Registers (DMATCR) 313
10.3.4 DMA Channel Control Registers (CHCR) 314
10.3.5 DMA Reload Source Address Registers (RSAR) 322
10.3.6 DMA Reload Destination Address Registers (RDAR) 323
10.3.7 DMA Reload Transfer Count Registers (RDMATCR) 324
10.3.8 DMA Operation Register (DMAOR) 325
10.3.9 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3) 329
10.4 Operation 331
10.4.1 Transfer Flow 331
10.4.2 DMA Transfer Requests 333
10.4.3 Channel Priority 337
10.4.4 DMA Transfer Types 340
10.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing 349
10.5 Interrupt Sources 353
10.5.1 Interrupt Sources and Priority Order 353
10.6 Usage Notes 355
10.6.1 Setting of the Half-End Flag and the Half-End Interrupt 355
10.6.2 Timing of DACK and TEND Outputs 355

Section 11 Multi-Function Timer Pulse Unit 2 (MTU2) 357
11.1 Features 357
11.2 Input/Output Pins 364
11.3 Register Descriptions 365
11.3.1 Timer Control Register (TCR) 369
11.3.2 Timer Mode Register (TMDR) 373
11.3.3 Timer I/O Control Register (TIOR) 376
11.3.4 Timer Compare Match Clear Register (TCNTCMPCLR) 395
11.3.5 Timer Interrupt Enable Register (TIER) 396
11.3.6 Timer Status Register (TSR) 401
11.3.7 Timer Buffer Operation Transfer Mode Register (TBTM) 408
11.3.8 Timer Input Capture Control Register (TICCR) 410
11.3.9 Timer Synchronous Clear Register S (TSYCRS) 411
11.3.10 Timer A/D Converter Start Request Control Register (TADCR) 413

11.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) ... 416
11.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) ... 416
11.3.13 Timer Counter (TCNT) ... 417
11.3.14 Timer General Register (TGR) ... 417
11.3.15 Timer Start Register (TSTR) ... 418
11.3.16 Timer Synchronous Register (TSYR) ... 420
11.3.17 Timer Counter Synchronous Start Register (TCSYSTR) ... 422
11.3.18 Timer Read/Write Enable Register (TRWER) ... 425
11.3.19 Timer Output Master Enable Register (TOER) ... 426
11.3.20 Timer Output Control Register 1 (TOCR1) ... 427
11.3.21 Timer Output Control Register 2 (TOCR2) ... 430
11.3.22 Timer Output Level Buffer Register (TOLBR) ... 433
11.3.23 Timer Gate Control Register (TGCR) ... 434
11.3.24 Timer Subcounter (TCNTS) ... 436
11.3.25 Timer Dead Time Data Register (TDDR) ... 437
11.3.26 Timer Cycle Data Register (TCDR) ... 437
11.3.27 Timer Cycle Buffer Register (TCBR) ... 438
11.3.28 Timer Interrupt Skipping Set Register (TITCR) ... 438
11.3.29 Timer Interrupt Skipping Counter (TITCNT) ... 440
11.3.30 Timer Buffer Transfer Set Register (TBTER) ... 441
11.3.31 Timer Dead Time Enable Register (TDER) ... 443
11.3.32 Timer Waveform Control Register (TWCR) ... 444
11.3.33 Bus Master Interface ... 446
11.4 Operation ... 447
11.4.1 Basic Functions ... 447
11.4.2 Synchronous Operation ... 453
11.4.3 Buffer Operation ... 455
11.4.4 Cascaded Operation ... 459
11.4.5 PWM Modes ... 464
11.4.6 Phase Counting Mode ... 469
11.4.7 Reset-Synchronized PWM Mode ... 476
11.4.8 Complementary PWM Mode ... 479
11.4.9 A/D Converter Start Request Delaying Function ... 525
11.4.10 MTU2-MTU2S Synchronous Operation ... 530
11.4.11 External Pulse Width Measurement ... 534
11.4.12 Dead Time Compensation ... 535
11.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operation ... 538
11.5 Interrupt Sources ... 539

11.5.1 Interrupt Sources and Priorities 539
11.5.2 DMAC and DTC Activation 541
11.5.3 A/D Converter Activation 542
11.6 Operation Timing 544
11.6.1 Input/Output Timing 544
11.6.2 Interrupt Signal Timing 551
11.7 Usage Notes 557
11.7.1 Module Standby Mode Setting 557
11.7.2 Input Clock Restrictions 557
11.7.3 Caution on Period Setting 558
11.7.4 Contention between TCNT Write and Clear Operations 558
11.7.5 Contention between TCNT Write and Increment Operations 559
11.7.6 Contention between TGR Write and Compare Match 560
11.7.7 Contention between Buffer Register Write and Compare Match 561
11.7.8 Contention between Buffer Register Write and TCNT Clear 562
11.7.9 Contention between TGR Read and Input Capture 563
11.7.10 Contention between TGR Write and Input Capture 564
11.7.11 Contention between Buffer Register Write and Input Capture 565
11.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection .. 565
11.7.13 Counter Value during Complementary PWM Mode Stop 567
11.7.14 Buffer Operation Setting in Complementary PWM Mode 567
11.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag 568
11.7.16 Overflow Flags in Reset Synchronous PWM Mode 569
11.7.17 Contention between Overflow/Underflow and Counter Clearing 570
11.7.18 Contention between TCNT Write and Overflow/Underflow 571
11.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode 571
11.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 572
11.7.21 Interrupts in Module Standby Mode 572
11.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection 572
11.8 MTU2 Output Pin Initialization 573
11.8.1 Operating Modes 573
11.8.2 Reset Start Operation 573
11.8.3 Operation in Case of Re-Setting Due to Error during Operation, etc. 574
11.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. 575

Section 12 Multi-Function Timer Pulse Unit 2S (MTU2S) 605
12.1 Input/Output Pins 608

12.2 Register Descriptions 609

Section 13 Port Output Enable 2 (POE2) 613
13.1 Features 613
13.2 Input/Output Pins 615
13.3 Register Descriptions 617
13.3.1 Input Level Control/Status Register 1 (ICSR1) 618
13.3.2 Output Level Control/Status Register 1 (OCSR1) 620
13.3.3 Input Level Control/Status Register 2 (ICSR2) 621
13.3.4 Output Level Control/Status Register 2 (OCSR2) 622
13.3.5 Input Level Control/Status Register 3 (ICSR3) 624
13.3.6 Software Port Output Enable Register (SPOER) 626
13.3.7 Port Output Enable Control Register 1 (POECR1) 627
13.3.8 Port Output Enable Control Register 2 (POECR2) 629
13.3.9 Port Output Enable Control Register 3 (POECR3) 634
13.4 Operation 636
13.4.1 Input Level Detection Operation 639
13.4.2 Output-Level Compare Operation 640
13.4.3 Release from High-Impedance State 641
13.5 Interrupts 642
13.6 Usage Notes 643
13.6.1 Pins States when the Watchdog Timer has Issued a Power-on Reset 643
13.6.2 Input Pins 643

Section 14 Compare Match Timer (CMT) 645
14.1 Features 645
14.2 Register Descriptions 646
14.2.1 Compare Match Timer Start Register (CMSTR) 647
14.2.2 Compare Match Timer Control/Status Register (CMCSR) 648
14.2.3 Compare Match Counter (CMCNT) 650
14.2.4 Compare Match Constant Register (CMCOR) 650
14.3 Operation 651
14.3.1 Interval Count Operation 651
14.3.2 CMCNT Count Timing 651
14.4 Interrupts 652
14.4.1 Interrupt Sources and DTC/DMAC Transfer Requests 652
14.4.2 Timing of Compare Match Flag Setting 653
14.4.3 Timing of Compare Match Flag Clearing 653
14.5 Usage Notes 654
14.5.1 Conflict between Write and Compare-Match Processes of CMCNT 654

14.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT 655
14.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT 656
14.5.4 Compare Match between CMCNT and CMCOR 656

Section 15 Watchdog Timer (WDT) 657
15.1 Features 657
15.2 Input/Output Pin 658
15.3 Register Descriptions 659
15.3.1 Watchdog Timer Counter (WTCNT) 659
15.3.2 Watchdog Timer Control/Status Register (WTCSR) 660
15.3.3 Watchdog Reset Control/Status Register (WRCSR) 662
15.3.4 Notes on Register Access 663
15.4 WDT Usage 665
15.4.1 Canceling Software Standby Mode 665
15.4.2 Using Watchdog Timer Mode 665
15.4.3 Using Interval Timer Mode 667
15.5 Interrupt Source 668
15.6 Usage Notes 668
15.6.1 Timer Variation 668
15.6.2 Prohibition against Setting H'FF to WTCNT 668
15.6.3 System Reset by $\overline{\text{WDTOVF}}$ Signal 669
15.6.4 Manual Reset in Watchdog Timer Mode 669
15.6.5 Connection of the $\overline{\text{WDTOVF}}$ Pin 669

Section 16 Serial Communication Interface (SCI) 671
16.1 Features 671
16.2 Input/Output Pins 673
16.3 Register Descriptions 674
16.3.1 Receive Shift Register (SCRSR) 676
16.3.2 Receive Data Register (SCRDR) 676
16.3.3 Transmit Shift Register (SCTSR) 676
16.3.4 Transmit Data Register (SCTDR) 677
16.3.5 Serial Mode Register (SCSMR) 677
16.3.6 Serial Control Register (SCSCR) 681
16.3.7 Serial Status Register (SCSSR) 684
16.3.8 Serial Port Register (SCSPTR) 690
16.3.9 Serial Direction Control Register (SCSDCR) 692
16.3.10 Bit Rate Register (SCBRR) 693
16.3.11 Sampling Mode Register (SPMR) 704
16.4 Operation 706

16.4.1 Overview 706
16.4.2 Operation in Asynchronous Mode 708
16.4.3 Clock Synchronous Mode 719
16.4.4 Multiprocessor Communication Function 728
16.4.5 Multiprocessor Serial Data Transmission 730
16.4.6 Multiprocessor Serial Data Reception 731
16.5 SCI Interrupt Sources and DTC 734
16.6 Serial Port Register (SCSPTR) and SCI Pins 735
16.7 Usage Notes 737
16.7.1 SCTDR Writing and TDRE Flag 737
16.7.2 Multiple Receive Error Occurrence 737
16.7.3 Break Detection and Processing 738
16.7.4 Sending a Break Signal 738
16.7.5 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) 738
16.7.6 Note on Using DTC 740
16.7.7 Note on Using External Clock in Clock Synchronous Mode 740
16.7.8 Module Standby Mode Setting 740
16.7.9 Note for RXD Pin State on Setting RE Bit 740
16.7.10 Clearing Interrupt Flags 741

Section 17 Serial Communication Interface with FIFO (SCIF) 743
17.1 Features 743
17.2 Input/Output Pins 745
17.3 Register Descriptions 745
17.3.1 Receive Shift Register (SCRSR) 746
17.3.2 Receive FIFO Data Register (SCFRDR) 746
17.3.3 Transmit Shift Register (SCTSR) 747
17.3.4 Transmit FIFO Data Register (SCFTDR) 747
17.3.5 Serial Mode Register (SCSMR) 748
17.3.6 Serial Control Register (SCSCR) 751
17.3.7 Serial Status Register (SCFSR) 755
17.3.8 Bit Rate Register (SCBRR) 763
17.3.9 FIFO Control Register (SCFCR) 771
17.3.10 FIFO Data Count Register (SCFDR) 773
17.3.11 Serial Port Register (SCSPTR) 774
17.3.12 Line Status Register (SCLSR) 775
17.3.13 Serial Extended Mode Register (SCSEMR) 777
17.4 Operation 778
17.4.1 Overview 778

17.4.2 Operation in Asynchronous Mode 780
17.4.3 Operation in Clocked Synchronous Mode 790
17.5 SCIF Interrupts 799
17.6 Usage Notes 800
17.6.1 SCFTDR Writing and TDFE Flag 800
17.6.2 SCFRDR Reading and RDF Flag 800
17.6.3 Restriction on DMAC and DTC Usage 801
17.6.4 Break Detection and Processing 801
17.6.5 Sending a Break Signal 801
17.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) 802
17.6.7 FER Flag and PER Flag of Serial Status Register (SCFSR) 803

Section 18 Renesas Serial Peripheral Interface (RSPI) 805
18.1 Features 805
18.2 Input/Output Pins 809
18.3 Register Descriptions 810
18.3.1 RSPI Control Register (SPCR) 811
18.3.2 RSPI Slave Select Polarity Register (SSLP) 814
18.3.3 RSPI Pin Control Register (SPPCR) 815
18.3.4 RSPI Status Register (SPSR) 816
18.3.5 RSPI Data Register (SPDR) 821
18.3.6 RSPI Sequence Control Register (SPSCR) 822
18.3.7 RSPI Sequence Status Register (SPSSR) 823
18.3.8 RSPI Bit Rate Register (SPBR) 825
18.3.9 RSPI Data Control Register (SPDCR) 826
18.3.10 RSPI Clock Delay Register (SPCKD) 831
18.3.11 SPI Slave Select Negation Delay Register (SSLND) 832
18.3.12 RSPI Next-Access Delay Register (SPND) 833
18.3.13 RSPI Command Register (SPCMD) 834
18.4 Operation 839
18.4.1 Overview of RSPI Operations 839
18.4.2 Controlling RSPI Pins 841
18.4.3 RSPI System Configuration Example 843
18.4.4 Transfer Format 852
18.4.5 Data Format 855
18.4.6 Transmit Buffer Empty/Receive Buffer Full Flags 861
18.4.7 Error Detection 863
18.4.8 Initializing RSPI 868
18.4.9 SPI Operation 869

18.4.10 Clock Synchronous Operation 881
18.4.11 Error Processing 888
18.4.12 Loopback Mode 890
18.4.13 Interrupt Request 891
18.5 Usage Notes 892
18.5.1 DTC Block Transfer 892
18.5.2 DMAC Burst Transfer 892
18.5.3 Reading Receive Data 892
18.5.4 DTC/DMAC and Mode Fault Error 892
18.5.5 Usage of the RSPI Output Pins as Open Drain Outputs 892
18.5.6 Unused Pins in Slave Mode 893

Section 19 A/D Converter (ADC) 895
19.1 Features 895
19.2 Input/Output Pins 898
19.3 Register Descriptions 899
19.3.1 A/D Control Registers 0 to 2 (ADCR_0 to ADCR_2) 901
19.3.2 A/D Status Registers 0 to 2 (ADSR_0 to ADSR_2) 904
19.3.3 A/D Start Trigger Select Registers 0 to 2 (ADSTRGR_0 to ADSTRGR_2) 905
19.3.4 A/D Analog Input Channel Select Registers 0 to 2 (ADANSR_0 to ADANSR_2) 907
19.3.5 A/D Bypass Control Registers 0 to 2 (ADBYPSCR_0 to ADBYPSCR_2) 908
19.3.6 A/D Data Registers 0 to 15 (ADDR0 to ADDR7) 910
19.3.7 A/D Trigger Select Registers 0 to 2 (ADTSR_0 to ADTSR_2) 912
19.3.8 A/D Group-0 Data-0 Registers 0 to 2 (ADDR0GR0_0 to ADDR0GR0_2) 917
19.3.9 A/D Group-1 Data-2 Registers 0 to 2 (ADDR2GR1_0 to ADDR2GR1_2) 918
19.4 Operation 919
19.4.1 Single-Cycle Scan Mode 920
19.4.2 Continuous Scan Mode 923
19.4.3 2-Channel Scan Mode 926
19.4.4 Input Sampling and A/D Conversion Time 929
19.4.5 A/D Converter Activation by MTU2 and MTU2S 931
19.4.6 External Trigger Input Timing 932
19.4.7 Example of ADDR Auto-Clear Function 933
19.4.8 A/D Conversion Synchronization Function 935
19.5 Interrupt Sources and DMAC or DTC Transfer Requests 945
19.6 Definitions of A/D Conversion Accuracy 946
19.7 Usage Notes 948
19.7.1 Analog Input Voltage Range 948
19.7.2 Relationship between AVCC, AVSS and VCC, VSS 948

19.7.3 Range of AVREF Pin Settings 948
19.7.4 Notes on Board Design 948
19.7.5 Notes on Noise Countermeasures 949
19.7.6 Permissible Signal Source Impedance 949
19.7.7 Influences on Absolute Precision 950
19.7.8 Notes when Two or More A/D Modules Run Simultaneously 950

Section 20 Controller Area Network (RCAN-ET) 955
20.1 Summary 955
20.1.1 Overview 955
20.1.2 Scope 955
20.1.3 Audience 955
20.1.4 References 956
20.1.5 Features 956
20.2 Architecture 957
20.3 Programming Model - Overview 959
20.3.1 Memory Map 959
20.3.2 Mailbox Structure 960
20.3.3 RCAN-ET Control Registers 967
20.3.4 RCAN-ET Mailbox Registers 987
20.4 Application Note 998
20.4.1 Test Mode Settings 998
20.4.2 Configuration of RCAN-ET 999
20.4.3 Message Transmission Sequence 1005
20.4.4 Message Receive Sequence 1008
20.4.5 Reconfiguration of Mailbox 1010
20.5 Interrupt Sources 1012
20.6 DTC Interface 1013
20.7 DMAC Interface 1014
20.8 CAN Bus Interface 1015

Section 21 Pin Function Controller (PFC) 1017
21.1 Register Descriptions 1023
21.1.1 Port A I/O Registers H and L (PAIORH and PAIORL) 1025
21.1.2 Port A Control Registers H1 and L1 to L4 (PACRH1 and PACRL1 to PACRL4) 1026
21.1.3 Port A Pull-Up MOS Control Registers H and L (PAPCRH and PAPCRL) 1032
21.1.4 Port B I/O Register H, L (PBIORH, PBIORL) 1034

21.1.5 Port B Control Register H1, H2, L1, and L2 (PBCRH1, PBCRH2, PBCRL1, and PBCRL2) ... 1035
21.1.6 Port B Pull-Up MOS Control Registers H and L (PBPCRH, PBPCRL) ... 1042
21.1.7 Port C I/O Register L (PCIORL) ... 1044
21.1.8 Port C Control Registers L1 to L4 (PCCRL1 to PCCRL4) ... 1044
21.1.9 Port C Pull-Up MOS Control Register L (PCPCRL) ... 1053
21.1.10 Port D I/O Register L (PDIORL) ... 1054
21.1.11 Port D Control Registers L1 to L4 (PDCRL1 to PDCRL4) ... 1055
21.1.12 Port D Pull-Up MOS Control Register L (PDPCRL) ... 1063
21.1.13 Port E I/O Register L (PEIORL) ... 1064
21.1.14 Port E Control Registers L1 to L4 (PECRL1 to PECRL4) ... 1065
21.1.15 Port E Pull-Up MOS Control Register L (PEPCRL) ... 1073
21.1.16 Large Current Port Control Register (HCPCR) ... 1074
21.1.17 DACK Output Timing Control Register (PDACKCR) ... 1075
21.2 Pull-Up MOS Control by Pin Function ... 1080
21.3 Usage Notes ... 1084

Section 22 I/O Ports ... 1085
22.1 Port A ... 1086
22.1.1 Register Descriptions ... 1086
22.1.2 Port A Data Registers H and L (PADRH and PADRL) ... 1087
22.1.3 Port A Port Registers H and L (PAPRH and PAPRL) ... 1089
22.2 Port B ... 1091
22.2.1 Register Descriptions ... 1091
22.2.2 Port B Data Registers H and L (PBDRH and PBDRL) ... 1092
22.2.3 Port B Port Registers H and L (PBPRH and PBPRL) ... 1094
22.3 Port C ... 1096
22.3.1 Register Descriptions ... 1097
22.3.2 Port C Data Register L (PCDRL) ... 1097
22.3.3 Port C Port Register L (PCPRL) ... 1099
22.4 Port D ... 1100
22.4.1 Register Descriptions ... 1101
22.4.2 Port D Data Register L (PDDRL) ... 1101
22.4.3 Port D Port Register L (PDPRL) ... 1103
22.5 Port E ... 1105
22.5.1 Register Descriptions ... 1105
22.5.2 Port E Data Register L (PEDRL) ... 1106
22.5.3 Port E Port Register L (PEPRL) ... 1107
22.6 Port F ... 1109
22.6.1 Register Descriptions ... 1109

22.6.2 Port F Data Register L (PFDRL) ... 1109
22.7 Usage Notes ... 1111
22.7.1 Handling of Unused pins ... 1111

Section 23 Flash Memory (ROM) ... 1113
23.1 Features ... 1113
23.2 Input/Output Pins ... 1118
23.3 Register Descriptions ... 1119
23.3.1 Flash Pin Monitor Register (FPMON) ... 1120
23.3.2 Flash Mode Register (FMODR) ... 1121
23.3.3 Flash Access Status Register (FASTAT) ... 1122
23.3.4 Flash Access Error Interrupt Enable Register (FAEINT) ... 1124
23.3.5 ROM MAT Select Register (ROMMAT) ... 1125
23.3.6 FCU RAM Enable Register (FCURAME) ... 1126
23.3.7 Flash Status Register 0 (FSTATR0) ... 1127
23.3.8 Flash Status Register 1 (FSTATR1) ... 1131
23.3.9 Flash P/E Mode Entry Register (FENTRYR) ... 1132
23.3.10 Flash Protect Register (FPROTR) ... 1134
23.3.11 Flash Reset Register (FRESETR) ... 1135
23.3.12 FCU Command Register (FCMDR) ... 1136
23.3.13 FCU Processing Switch Register (FCPSR) ... 1137
23.3.14 Flash P/E Status Register (FPESTAT) ... 1138
23.3.15 ROM Cache Control Register (RCCR) ... 1139
23.3.16 Peripheral Clock Notification Register (PCKAR) ... 1140
23.4 Overview of ROM-Related Modes ... 1141
23.5 Boot Mode ... 1143
23.5.1 System Configuration ... 1143
23.5.2 State Transition in Boot Mode ... 1144
23.5.3 Automatic Adjustment of Bit Rate ... 1146
23.5.4 Inquiry/Selection Host Command Wait State ... 1147
23.5.5 Programming/Erasing Host Command Wait State ... 1165
23.6 User Program Mode ... 1177
23.6.1 FCU Command List ... 1177
23.6.2 Conditions for FCU Command Acceptance ... 1180
23.6.3 FCU Command Usage ... 1184
23.6.4 Suspending Operation ... 1203
23.7 User Boot Mode ... 1206
23.7.1 User Boot Mode Initiation ... 1206
23.7.2 User MAT Programming ... 1208
23.8 Programmer Mode ... 1209

23.9 Protection ... 1209
23.9.1 Hardware Protection ... 1209
23.9.2 Software Protection ... 1210
23.9.3 Error Protection ... 1211
23.10 Usage Notes ... 1214
23.10.1 Switching between User MAT and User Boot MAT ... 1214
23.10.2 State in which Interrupts are Ignored ... 1216
23.10.3 Programming-/Erasure-Suspended Area ... 1216
23.10.4 Compatibility with Programming/ Erasing Program of Conventional F-ZTAT SH Microcomputers ... 1216
23.10.5 FWE Pin State ... 1216
23.10.6 Reset during Programming or Erasure ... 1217
23.10.7 Suspension by Programming/Erasure Suspension ... 1217
23.10.8 Prohibition of Additional Programming ... 1218
23.10.9 Allocation of Interrupt Vectors during Programming and Erasure ... 1218
23.10.10 Items Prohibited during Programming and Erasure ... 1218

Section 24 Data Flash (FLD) ... 1219
24.1 Features ... 1219
24.2 Input/Output Pins ... 1224
24.3 Register Descriptions ... 1224
24.3.1 Flash Mode Register (FMODR) ... 1226
24.3.2 Flash Access Status Register (FASTAT) ... 1227
24.3.3 Flash Access Error Interrupt Enable Register (FAEINT) ... 1230
24.3.4 FLD Read Enable Register 0 (EEPRE0) ... 1232
24.3.5 FLD Read Enable Register 1 (EEPRE1) ... 1233
24.3.6 FLD Program/Erase Enable Register 0 (EEPWE0) ... 1234
24.3.7 FLD Program/Erase Enable Register 1 (EEPWE1) ... 1235
24.3.8 Flash P/E Mode Entry Register (FENTRYR) ... 1236
24.3.9 FLD Blank Check Register (EEPBCCNT) ... 1238
24.3.10 FLD Blank Check Status Register (EEPBCSTAT) ... 1239
24.4 Overview of FLD-Related Modes ... 1240
24.5 Boot Mode ... 1242
24.5.1 Inquiry/Selection Host Commands ... 1242
24.5.2 Programming/Erasing Host Commands ... 1245
24.6 User Mode, User Program Mode, and User Boot Mode ... 1247
24.6.1 FCU Command List ... 1247
24.6.2 Conditions for FCU Command Acceptance ... 1249
24.6.3 FCU Command Usage ... 1253
24.7 Protection ... 1258

24.7.1 Hardware Protection 1258
24.7.2 Software Protection 1258
24.7.3 Error Protection 1259
24.8 Usage Notes 1261
24.8.1 Protection of Data MAT Immediately after a Reset 1261
24.8.2 State in which Interrupts are Ignored 1261
24.8.3 Programming-/Erasure-Suspended Area 1261
24.8.4 Compatibility with Programming/ Erasing Program of Conventional F-ZTAT SH Microcontrollers 1261
24.8.5 Reset during Programming or Erasure 1262
24.8.6 Suspension by Programming/Erasure Suspension 1262
24.8.7 Prohibition of Additional Programming 1262
24.8.8 Program for Reading 1262
24.8.9 Items Prohibited during Programming and Erasure 1263

Section 25 On-Chip RAM 1265
25.1 Features 1265
25.2 Register Descriptions 1267
25.2.1 System Control Register 1 (SYSCR1) 1268
25.2.2 System Control Register 2 (SYSCR2) 1270
25.3 Notes on Usage 1272
25.3.1 Page Conflict 1272

Section 26 Power-Down Modes 1273
26.1 Features 1273
26.1.1 Power-Down Modes 1273
26.1.2 Reset 1274
26.2 Input/Output Pins 1275
26.3 Register Descriptions 1276
26.3.1 Standby Control Register (STBCR) 1276
26.3.2 Standby Control Register 2 (STBCR2) 1277
26.3.3 Standby Control Register 3 (STBCR3) 1278
26.3.4 Standby Control Register 4 (STBCR4) 1280
26.3.5 Standby Control Register 5 (STBCR5) 1281
26.3.6 Standby Control Register 6 (STBCR6) 1283
26.4 Operation 1284
26.4.1 Sleep Mode 1284
26.4.2 Software Standby Mode 1284
26.4.3 Application Example of Software Standy Mode 1287
26.4.4 Module Standby Function 1288

Section 27 User Debugging Interface (H-UDI) 1289
27.1 Features 1289
27.2 Input/Output Pins 1290
27.3 Register Descriptions 1291
27.3.1 Bypass Register (SDBPR) 1291
27.3.2 Instruction Register (SDIR) 1292
27.4 Operation 1293
27.4.1 TAP Controller 1293
27.4.2 Reset Configuration 1294
27.4.3 TDO Output Timing 1294
27.4.4 H-UDI Reset 1295
27.4.5 H-UDI Interrupt 1295
27.5 Usage Notes 1296

Section 28 List of Registers 1297
28.1 Register Addresses
(by Functional Module, in Order of the Corresponding Section Numbers) 1298
28.2 Register Bits 1323
28.3 Register States in Each Operating Mode 1356

Section 29 Electrical Characteristics 1377
29.1 Absolute Maximum Ratings 1377
29.2 DC Characteristics 1378
29.3 AC Characteristics 1385
29.3.1 Clock Timing 1386
29.3.2 Control Signal Timing 1389
29.3.3 Bus Timing (SH7239A and SH7237A only) 1393
29.3.4 UBC Trigger Timing 1400
29.3.5 DMAC Module Timing 1401
29.3.6 MTU2, MTU2S Module Timing 1402
29.3.7 POE2 Module Timing 1403
29.3.8 Watchdog Timer Timing 1404
29.3.9 SCI Module Timing 1404
29.3.10 SCIF Module Timing 1406
29.3.11 RSPI Timing 1408
29.3.12 Controller Area Network (RCAN-ET) Timing 1412
29.3.13 A/D Trigger Input Timing 1413
29.3.14 I/O Port Timing 1414
29.3.15 H-UDI Related Pin Timing 1415
29.3.16 AC Characteristics Measurement Conditions 1417

29.4 A/D Converter Characteristics 1418
29.5 Flash Memory Characteristics 1419
29.6 FLD Characteristics 1421
29.7 Usage Note 1422
29.7.1 Notes on Connecting Capacitors 1422

Appendix 1423
A. Pin States 1423
B. Package Dimensions 1430

Index 1431

Section 1 Overview

1.1 Features

SH7239 Group and SH7237 Group are single-chip RISC (Reduced Instruction Set Computer) microprocessors that integrate a Renesas original RISC CPU core with peripheral functions required for system configuration.

The CPU in the SH7239 and SH7237 Groups has a RISC-type 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. The SH7239 Group also includes the floating-point unit (FPU).

In addition, the SH7239 and SH7237 Groups include 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, a controller area network (RCAN-ET), and data flash (FLD).

The SH7239 and SH7237 Groups also provide 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 (SH7239 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 MCU extension mode 2 (SH7239A and SH7237A only) 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)	• Eight external interrupt pins (NMI and IRQ6 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) (SH7239A and SH7237A only)	• Address space divided into six areas (0, 1, 3 to 6), each a maximum of 2 Mbytes • External bus: 8 or 16 bits • The following features settable for each area independently Supports both big endian and little endian for data access Bus size (8 or 16 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)
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: SH7239A, SH7237A: CPU clock: Maximum 160 MHz Bus clock: Maximum 40 MHz Peripheral clock: Maximum 40 MHz AD clock: Maximum 40 MHz MTU clock: Maximum 80 MHz SH7239B, SH7237B: CPU clock: Maximum 100 MHz Bus clock: Maximum 50 MHz Peripheral clock: Maximum 50 MHz AD clock: Maximum 50 MHz MTU clock: Maximum 100 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 • Operating frequency Operating at 100 MHz max: SH7239B, SH7237B Operating at 80 MHz max: SH7239A, SH7237A
Multi-function timer pulse unit 2S (MTU2S)	• Subset of MTU2, included in channels 3 to 5 • Operating frequency Operating at 100 MHz max: SH7239B, SH7237B Operating at 80 MHz max: SH7239A, SH7237A
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 • High-impedance control of multiple groups of high-current pins with a single $\overline{\text{POE}}$ 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

Items	Specification
Serial communication interface (SCI)	• Three channels • Clocked synchronous or asynchronous mode selectable • Simultaneous transmission and reception (full-duplex communication) supported • Dedicated baud rate generator • Noise canceller (for asynchronous communication only)
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 • Used as the SCI during boot mode
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: 10 Mbps (SH7239A, SH7237A) or 12.5 Mbps (SH7239B, SH7237B) • 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)
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/O ports	• Input or output can be selected for each bit

Items	Specification
A/D converter	• Three modules • 12-bit resolution • 16 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
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
Advanced user debugger (AUD)	• Six 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	• 512 or 256 Kbytes
On-chip RAM	• 64 or 32 Kbytes • Number of pages: Products with 64-Kbyte RAM: Four pages (pages 0, 1, 4, and 5) Products with 32-Kbyte RAM: Two pages (pages 0 and 1)
Data flash (FLD)	• 32 Kbytes (2 Kbytes × 16 blocks) • Programmed in 8-byte units
Power supply voltage	• VCC: 4.5 to 5.5 V (SH7239B, SH7237B) 3.0 to 3.6 V (SH7239A, SH7237A) • AVCC: 4.5 to 5.5 V
Packages	• LQFP1616-120 (0.5 pitch): SH7239A, SH7239B, SH7237A, SH7237B

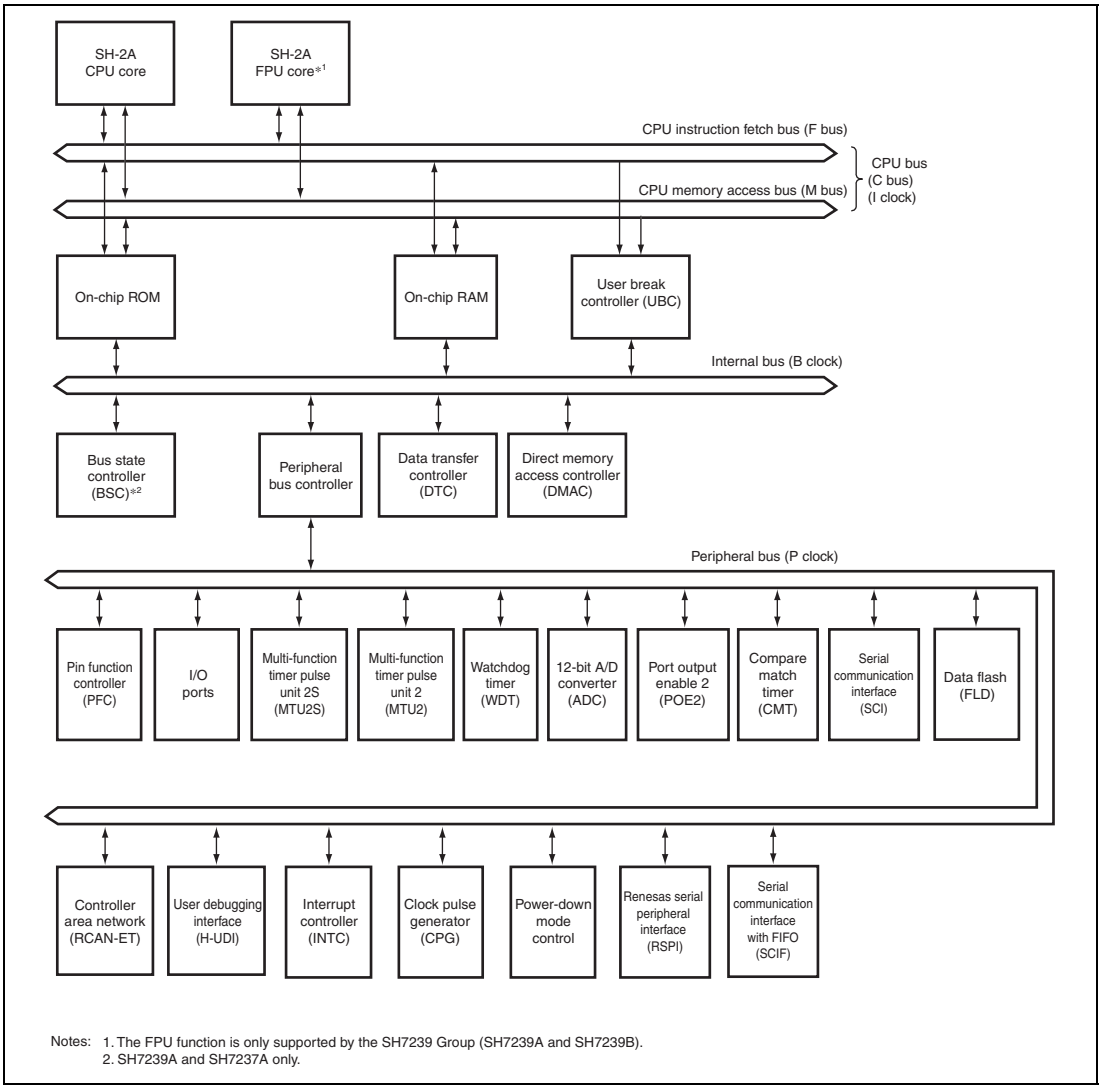
1.2 List of Products

Table 1.2 lists the products.

Table 1.2 Product Code Lineup

					Power Supply Voltage				
Group	Product	Part Name	ROM Capacity	RAM Capacity	VCC, PLLVCC	AVCC	Package	FPU Function	Extended Function
SH7239 Group	SH7239B	R5F72395BDFP	512 Kbytes	64 Kbytes	4.5 to 5.5 V	4.5 to 5.5 V	LQFP1616-120	Available	Unavailable
		R5F72394BDFP	256 Kbytes	32 Kbytes					
	SH7239A	R5F72395ADFP	512 Kbytes	64 Kbytes	3.0 to 3.6 V				Available
		R5F72394ADFP	256 Kbytes	32 Kbytes					
SH7237 Group	SH7237B	R5F72375BDFP	512 Kbytes	64 Kbytes	4.5 to 5.5 V	4.5 to 5.5 V	LQFP1616-120	Unavailable	Unavailable
		R5F72374BDFP	256 Kbytes	32 Kbytes					
	SH7237A	R5F72375ADFP	512 Kbytes	64 Kbytes	3.0 to 3.6 V				Available
		R5F72374ADFP	256 Kbytes	32 Kbytes					

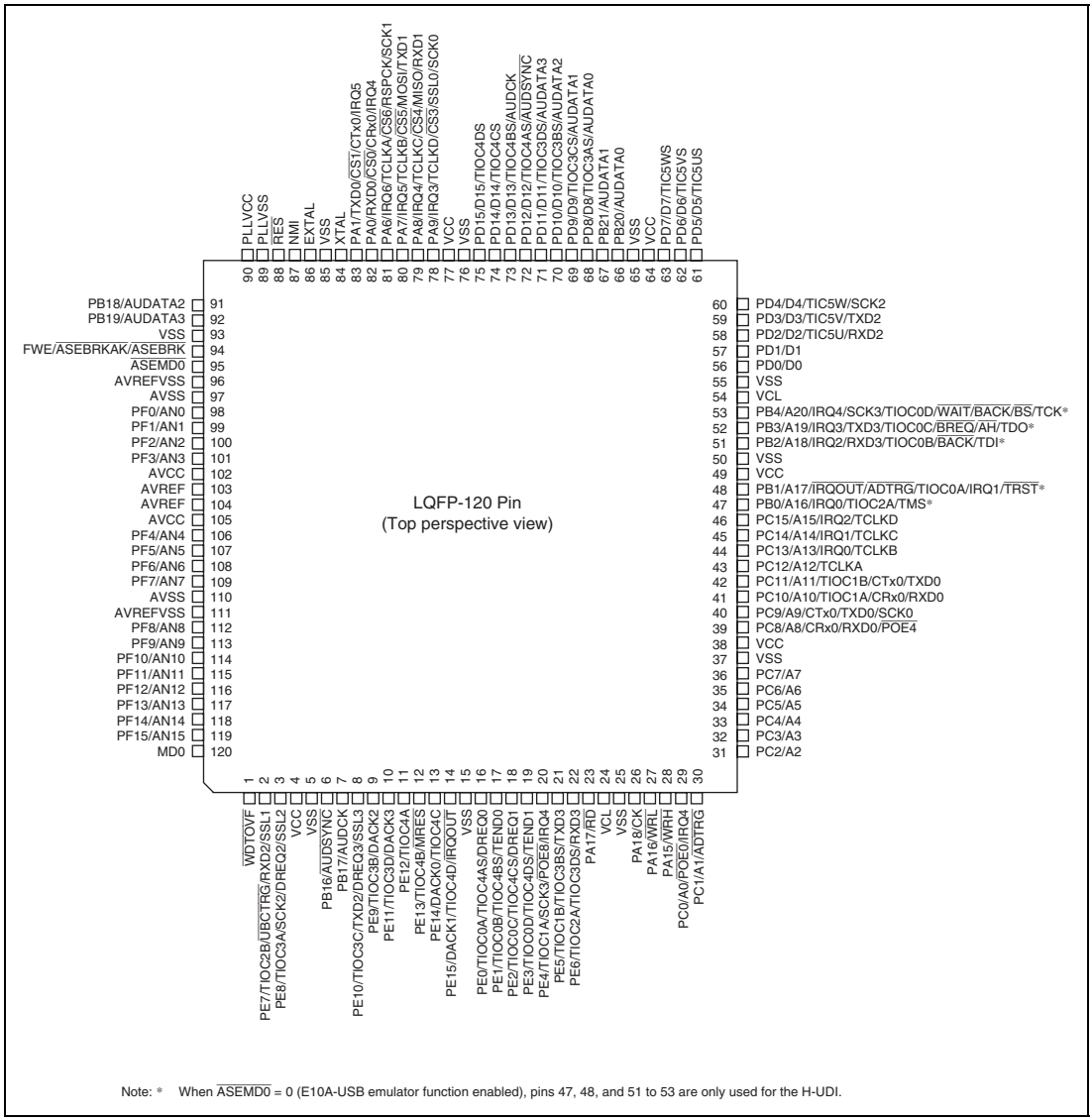
1.3 Block Diagram



Notes: 1. The FPU function is only supported by the SH7239 Group (SH7239A and SH7239B).
2. SH7239A and SH7237A only.

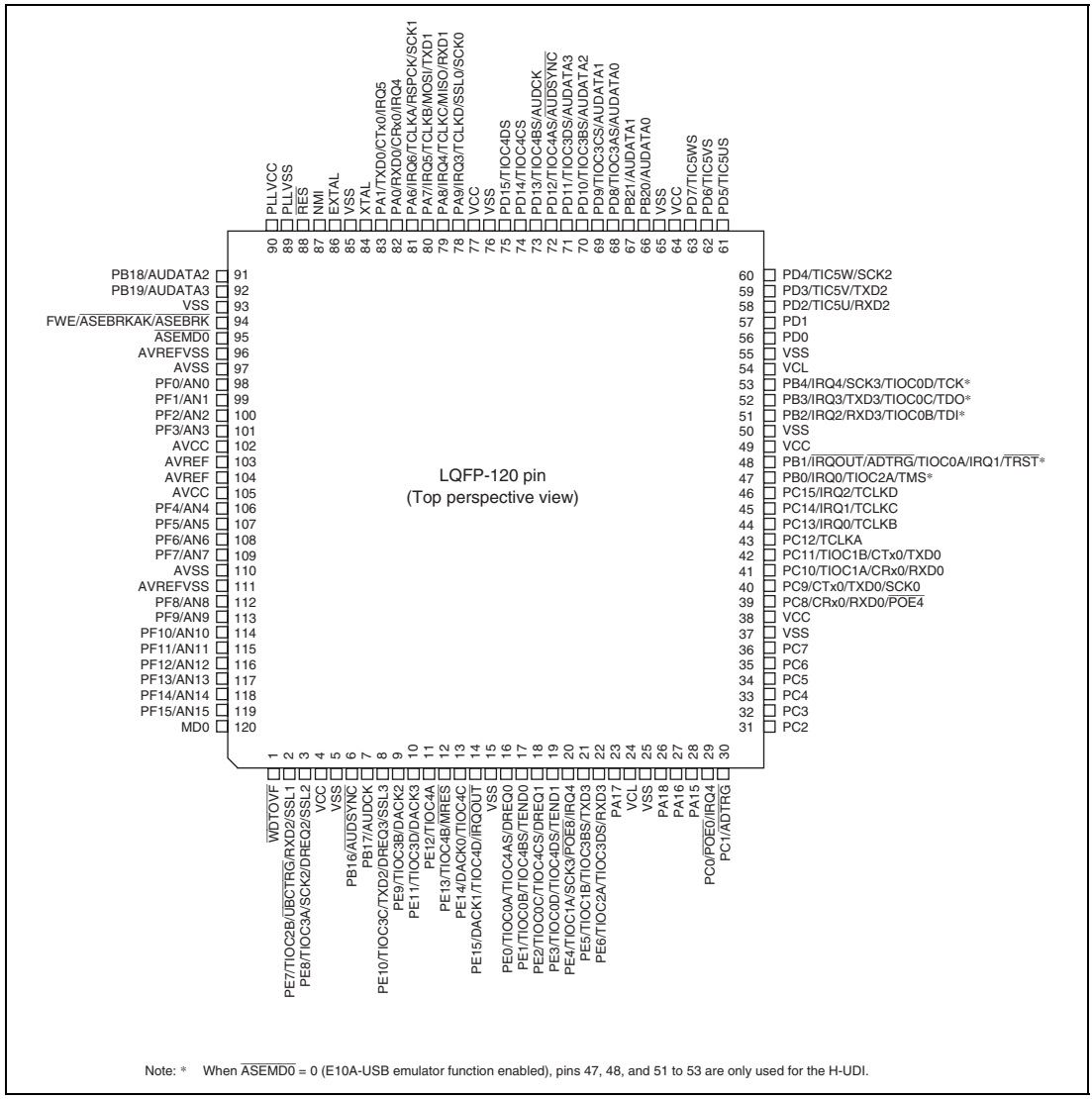
Figure 1.1 Block Diagram

1.4 Pin Assignment



Note: * When $\overline{\text{ASEMD0}}$ = 0 (E10A-USB emulator function enabled), pins 47, 48, and 51 to 53 are only used for the H-UDI.

Figure 1.2 Pin Assignment of SH7239A and SH7237A (120 Pins) (Top Perspective View)



Note: * When ASEMD0 = 0 (E10A-USB emulator function enabled), pins 47, 48, and 51 to 53 are only used for the H-UDI.

Figure 1.3 Pin Assignment of SH7239B and SH7237B (120 Pins) (Top Perspective View)

1.5 Pin Functions

Table 1.3 lists functions of each pin.

Table 1.3 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	VCL	Output	Internal step-down power supply	External capacitance pins for internal step-down power supply. All the VCL pins must be connected to the VSS 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 VCL pins.
	VSS	Input	Ground	Ground pins. All the VSS pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	VCC	Input	Power supply	Power supply pins. All the VCC pins must be connected to the system power supply. This LSI does not operate if there is a pin left open.
	PLLVCC	Input	PLL power supply	Power supply for the on-chip PLL oscillator. Apply the same electric potential as that on the VCC pin.
	PLLVSS	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.
	CK*[3]	Output	System clock	Supplies the system clock to external devices.

Classification	Symbol	I/O	Name	Function
Operating mode control	MD0	Input	Mode set	Sets the operating mode. Do not change the signal levels 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.
	$\overline{\text{BREQ}}$*2*3	Input	Bus-mastership request	A low level is input to this pin when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$*2*3	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.
	IRQ6, IRQ5, IRQ4 to IRQ0*1	Input	Interrupt requests 6 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.
	$\overline{\text{IRQOUT}}$*1	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*3	A20 to A16*2, A15 to A0	Output	Address bus	Outputs addresses.
Data bus*3	D15 to D0	I/O	Data bus	Bidirectional data bus.
Bus control*3	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS3}}$ to $\overline{\text{CS6}}$	Output	Chip select 0, 1, 3 to 6	Chip-select signals for external memory or devices.
	$\overline{\text{RD}}$	Output	Read	Indicates that data is read from an external device.
	$\overline{\text{BS}}$*2	Output	Bus start	Bus-cycle start signal.
	$\overline{\text{AH}}$*2	Output	Address hold	Address hold timing signal for the device that uses the address/data-multiplexed bus.
	$\overline{\text{WAIT}}$*2	Input	Wait	Input signal for inserting a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WRH}}$	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.
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*[1], TIOC0B*[1], TIOC0C*[1], TIOC0D*[1]	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*[1], 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)	$\overline{\text{POE8}}$, $\overline{\text{POE4}}$, $\overline{\text{POE0}}$	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)	TXD2 to TXD0	Output	Transmit data	Data output pins.
	RXD2 to RXD0	Input	Receive data	Data input pins.
	SCK2 to SCK0	I/O	Serial clock	Clock input/output pins.
Serial communication interface with FIFO (SCIF)	TXD3*[1]	Output	Transmit data	Data output pin.
	RXD3*[1]	Input	Receive data	Data input pin.
	SCK3*[1]	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
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.
A/D converter	AN15 to AN0	Input	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$*1	Input	A/D conversion trigger input	External trigger input pin for starting A/D conversion.
	AVCC	Input	Analog power supply	Power supply pin for the A/D converter. Connect this pin to the system power supply (VCC) when the A/D converter is not used.
	AVREF	Input	Analog reference power supply	Reference voltage pin for the A/D converter.
	AVSS	Input	Analog ground	Ground pin for the A/D converter. Connect this pin to the system power supply (VSS) when the A/D converter is not used.
	AVREFVSS	Input	Analog reference ground	Reference ground pin for the A/D converter. Connect this pin to the system power supply (VSS) when the A/D converter is not used.

Classification	Symbol	I/O	Name	Function
I/O ports	PA18 to PA15, PA9 to PA6, PA1, PA0	I/O	General port	Ten general input/output port pins.
	PB21 to PB16, PB4 to PB0*2	I/O	General port	Eleven general input/output port pins.
	PC15 to PC0	I/O	General port	Sixteen general input/output port pins.
	PD15 to PD0	I/O	General port	Sixteen general input/output port pins.
	PE15 to PE0	I/O	General port	Sixteen general input/output port pins.
	PF15 to PF0	Input	General port	Sixteen 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

Classification	Symbol	I/O	Name	Function
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.

Notes: 1. The pins which are function-multiplexed with the H-UDI function are used as the H-UDI dedicated pins when $\overline{\text{ASEMD0}}$ = 0 (E10A-USB emulator function enabled).
2. These pins are function-multiplexed with the H-UDI function and are used as the H-UDI dedicated pins when $\overline{\text{ASEMD0}}$ = 0 (E10A-USB emulator function enabled).
3. The external extension pins are only available for SH7239A and SH7237A.

Section 2 CPU

2.1 Data Format

Figure 2.1 shows the data format supported by the SH-2A/SH2A-FPU. The SH2A-FPU is only supported by the SH7239 Group.

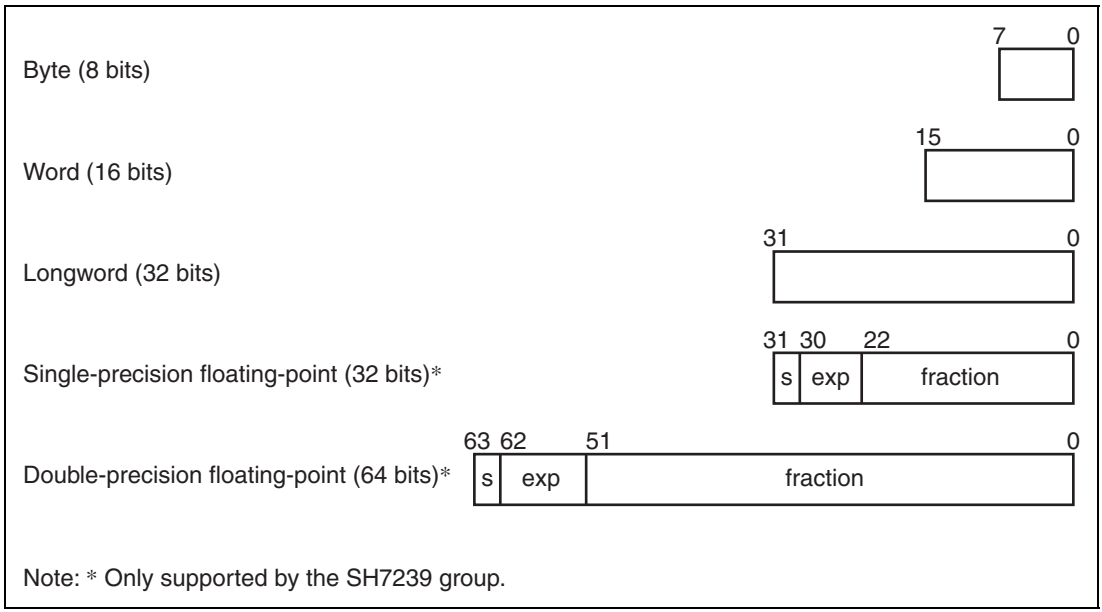


Figure 2.1 Data Format

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.

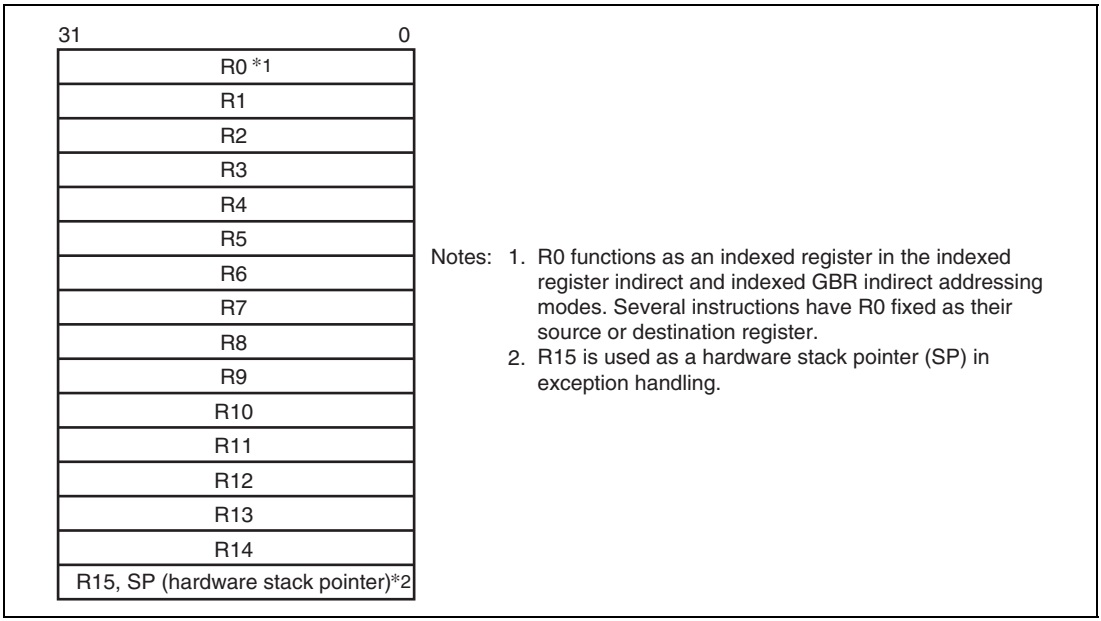


Figure 2.2 General Registers

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.

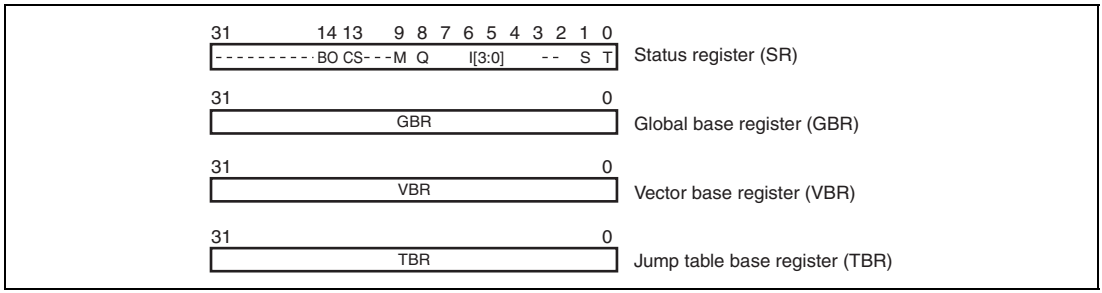


Figure 2.3 Control Registers

(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.

Bit	Bit Name	Initial Value	R/W	Description
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.

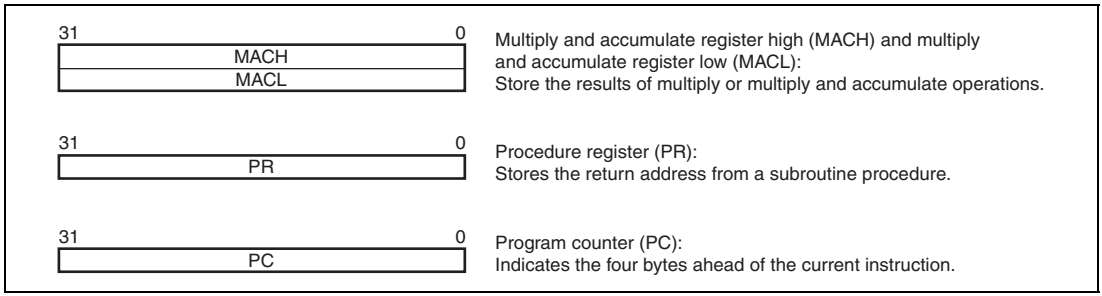


Figure 2.4 System Registers

(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 (SH7239 Group Only)

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}

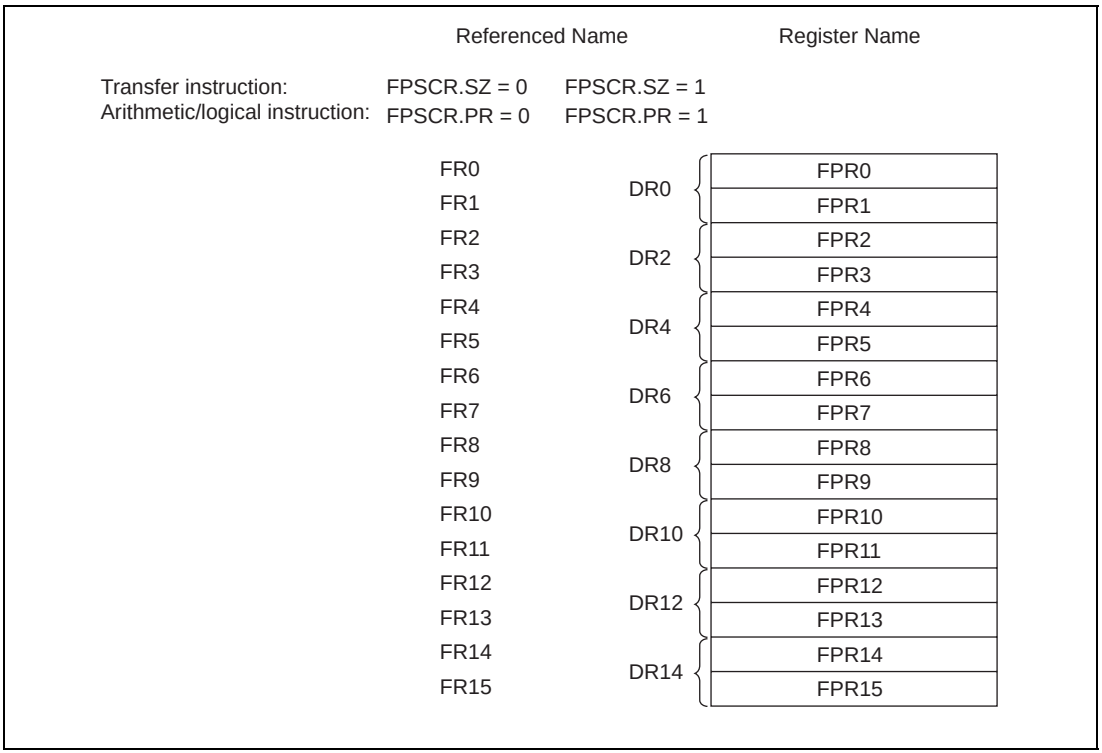


Figure 2.5 Floating-Point Registers

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

2.2.5 Floating-Point System Registers (SH7239 Group Only)

(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: The operation cannot be guaranteed in the SH7237 Group.

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 fifteen 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

Note: * These registers are only provided by the SH7239 Group. The operation cannot be guaranteed in the SH7237 Group.

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.

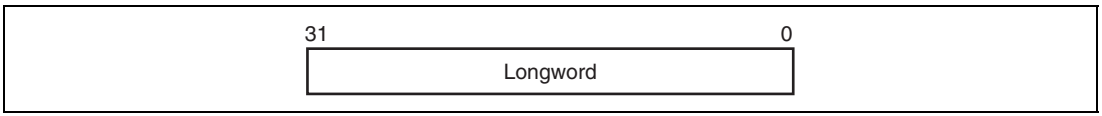


Figure 2.6 Data Format in Registers

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.

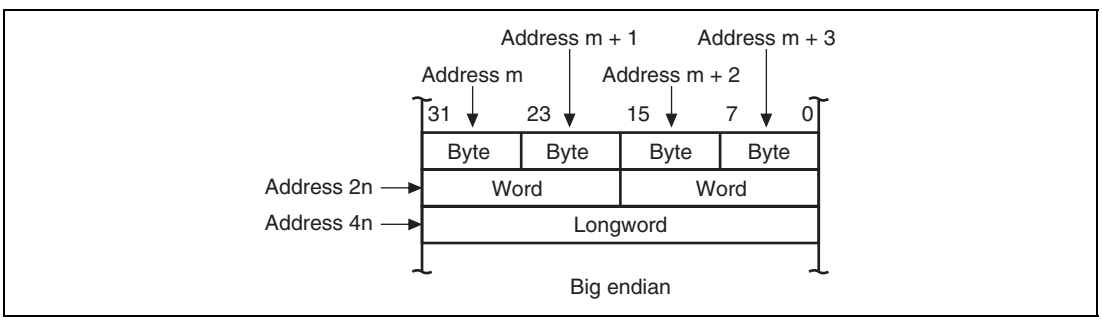


Figure 2.7 Data Formats in Memory

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.

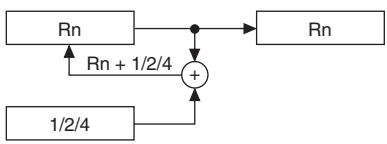
Table 2.8 Displacement Accessing

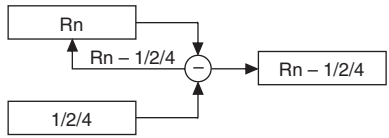
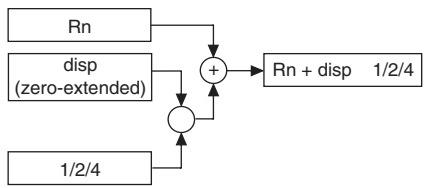
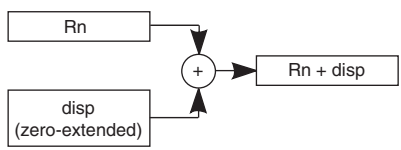
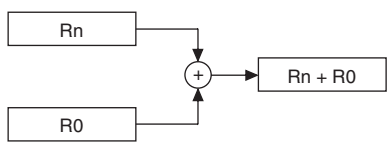
Classification	SH-2A/SH2A-FPU CPU		Example of Other CPU
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

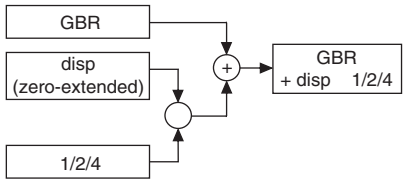
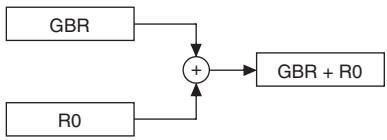
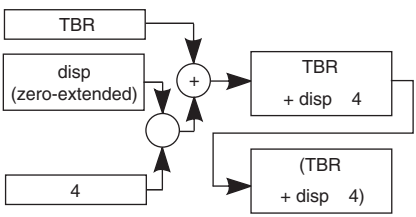
2.4.2 Addressing Modes

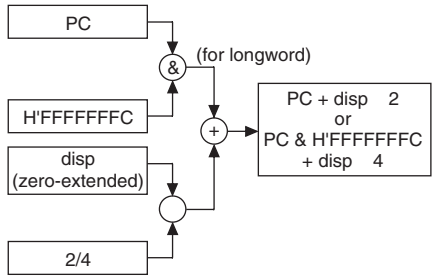
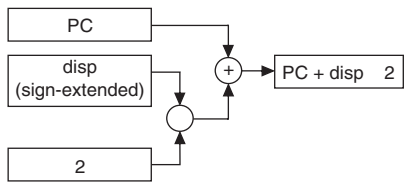
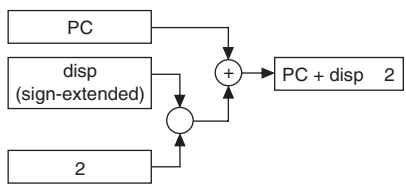
The addressing modes and effective address calculation methods are listed below.

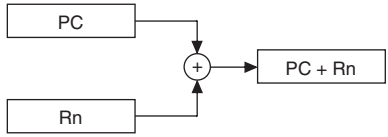
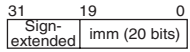
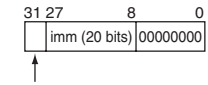
Table 2.9 Addressing Modes and Effective Addresses

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.	—

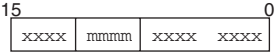
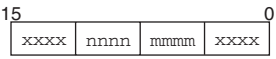
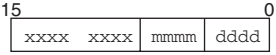
2.4.3 Instruction Format

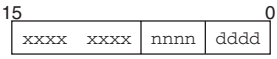
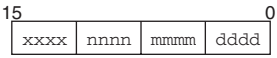
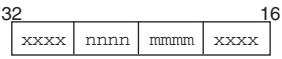
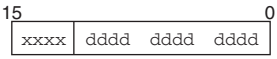
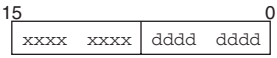
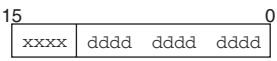
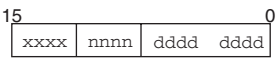
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

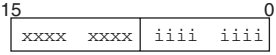
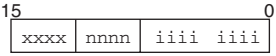
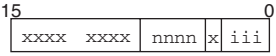
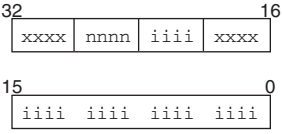
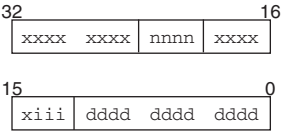
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

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)
	—	nnnnddddddddddd: 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	62
			Immediate data transfer	
			Peripheral module data transfer	
			Structure data transfer	
			Reverse stack transfer	
		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

Note: These registers are only provided by the SH7239 Group. The operation cannot be guaranteed in the SH7237 Group.

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	Compatibility SH4	Compatibility 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	Compatibility SH4	Compatibility 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 1000dddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W @(disp12, Rm), Rn	0011nnnnmmmm0001 1001dddddddddddd	(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	Overflow	Yes	Yes	
CMP/EQ	#imm, R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/EQ	Rm, Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/GE	Rm, Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/HI	Rm, Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	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	SH4	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	Com-parison 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	SH4	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	SH4	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 (SH7239 Group Only)

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	Comparison 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	SH4	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 (SH7239 Group Only)

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
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.

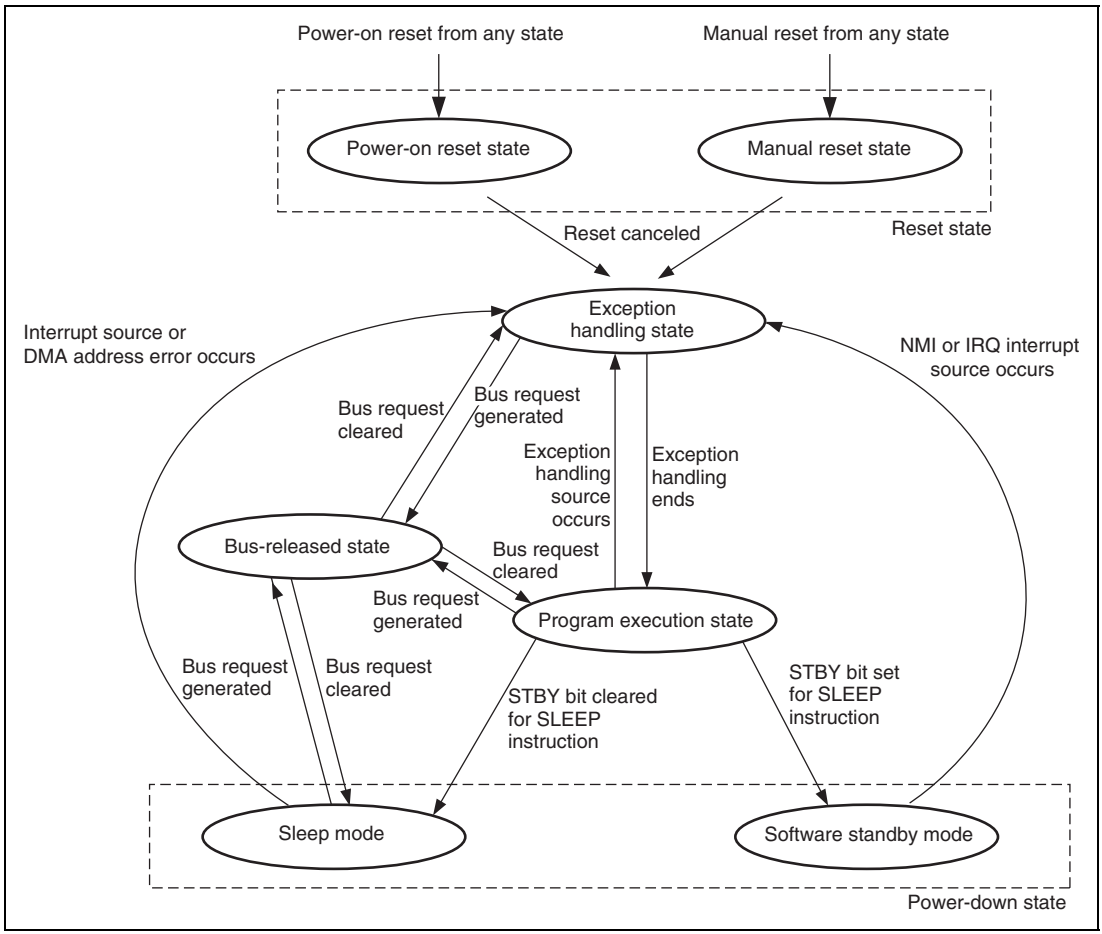


Figure 2.8 Transitions between Processing States

(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 two MCU operating modes and three on-chip flash memory programming modes. The operating mode is determined by the setting of FWE 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 mode 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 Operating Modes and Pin Settings

	Pin Setting				
Mode No.	**FWE**	**MD0**	**Mode Name**	**On-Chip ROM**	**Bus Width of CS0 Space**
Mode 2*5	0	0	MCU extension mode 2	Active	Set by CS0BCR in BSC
Mode 3	0	1	Single chip mode	Active	—
Mode 4*1*2	1	0	Boot mode	Active	—
Mode 4*1*3	1	0	User program mode	Active	Set by CS0BCR in BSC
Mode 6*1*2	1	1	User boot mode	Active	—
Mode 6*1*4	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 is set to 1 after the MCU operation has been set to MCU extension mode 2, transition is made to the user program mode in an MCU extension mode 2 state.
4. If FWE = 0 when power-on reset has been released, and if FWE is set 1 after the MCU operation has been set to single chip mode, transition to the user program mode is executed in a single chip mode state.
5. Mode 2 (MCU extension mode 2) and mode 4 (user program mode) are only available for the SH7239A and SH7237A.

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
FWE	Input	Enables, by hardware, programming/erasing of the on-chip flash memory

3.3 Operating Modes

3.3.1 Mode 2 (MCU Extension Mode 2)

The on-chip ROM is active and CS space can be used in this mode. This mode is only available for the SH7239A and SH7237A.

3.3.2 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 and 3.2.

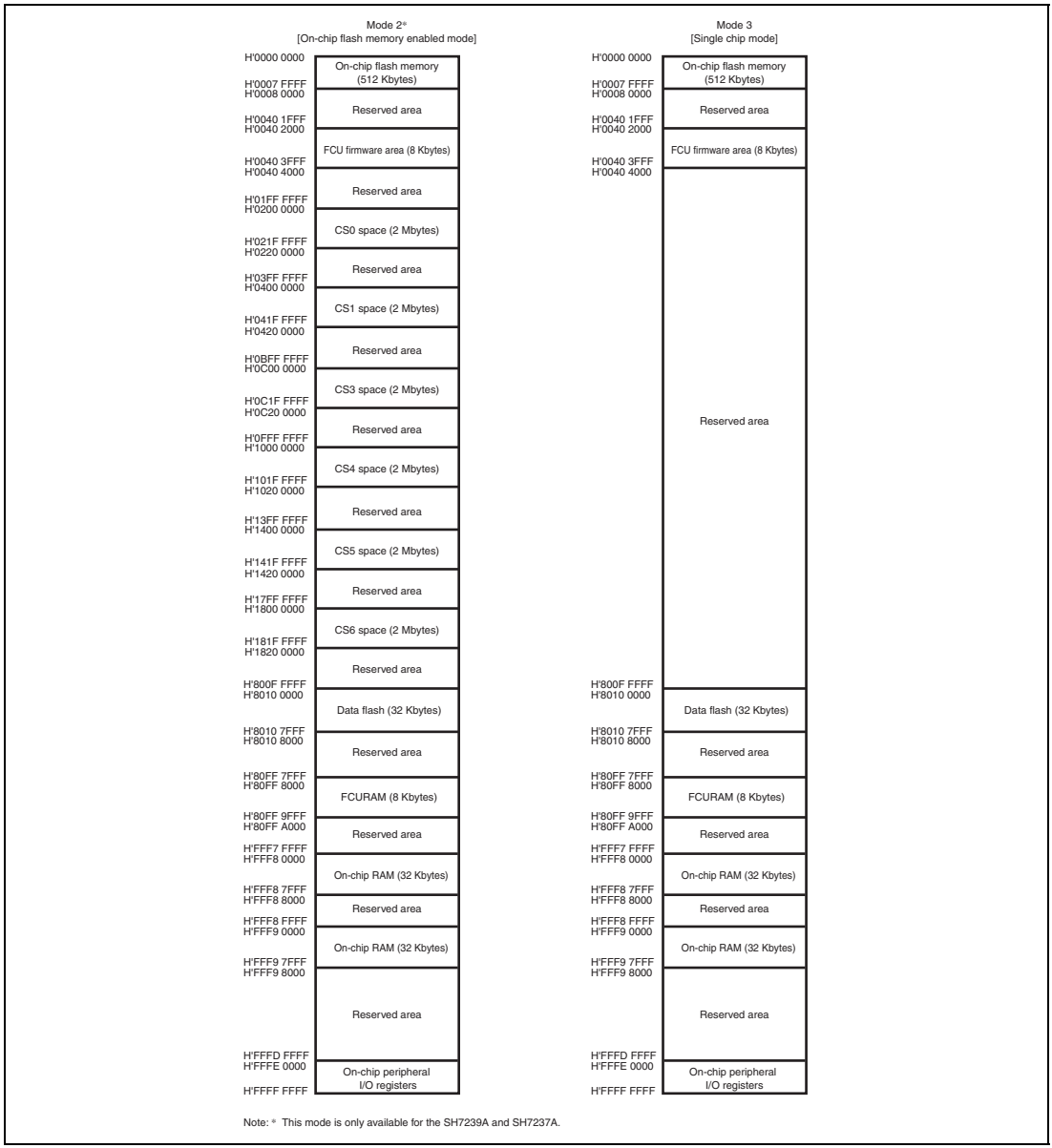
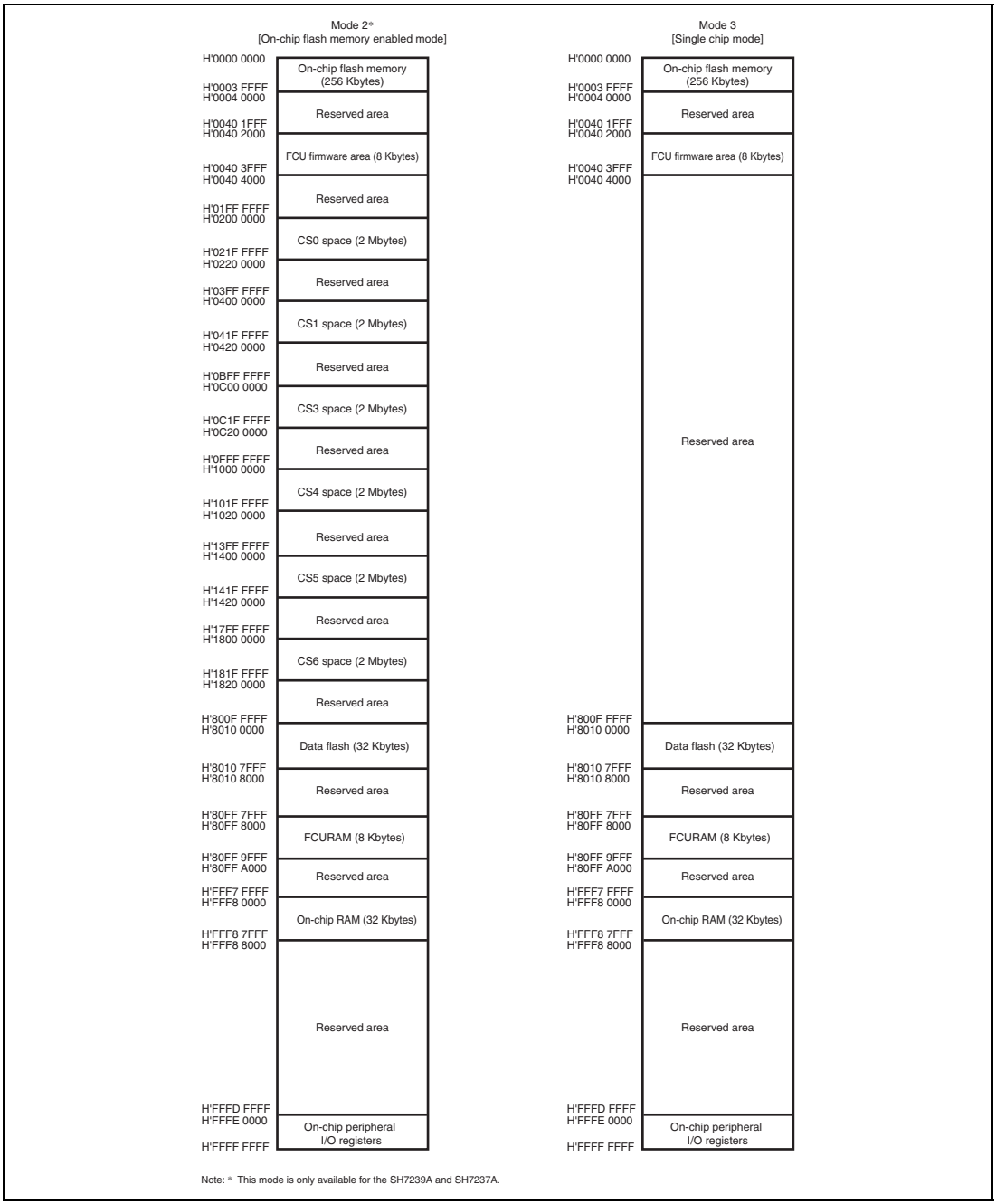


Figure 3.1 Address Map of Product with 512-Kbyte ROM



Note: * This mode is only available for the SH7239A and SH7237A.

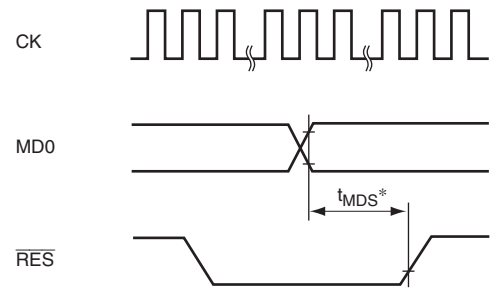
Figure 3.2 Address Map of Product with 256-Kbyte ROM

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 26, 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). However, this does not apply when the mode change is from mode 2 (MCU extension mode 2) to mode 4 (user program mode) or from mode 3 (single chip mode) to mode 6 (user program mode).



Note: * See section 29.3.2, Control Signal Timing.

Figure 3.3 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 MTU 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 ROM cache, a peripheral clock (Pϕ) for the peripheral modules, a bus clock (Bϕ = CK) for the external bus interface, an MTU clock (Mϕ) for the MTU2/MTU2S module, and an AD clock (Aϕ) for the A/D converter 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 26, Power-Down Modes.

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

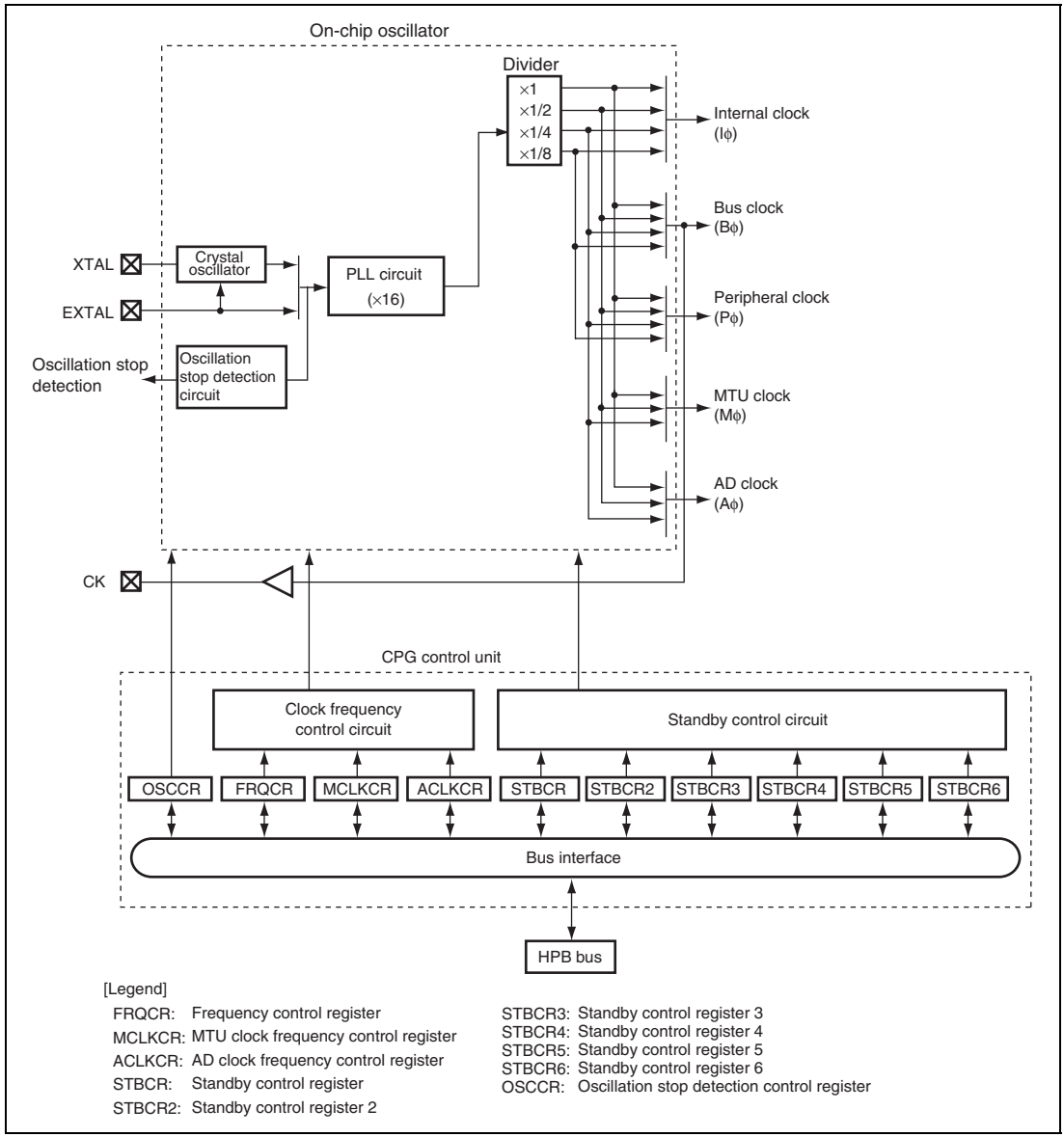


Figure 4.1 Block Diagram of Clock Pulse Generator

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), MTU 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).

(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) MTU Clock Frequency Control Register (MCLKCR)

The MTU clock frequency control register (MCLKCR) has control bits assigned for the following function: the frequency division ratio of the MTU 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. See section 26, 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.

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.

Note: * Can be used for the SH7239A and SH7237A only.

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 21, 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 of the SH7237 Group, appropriate settings should be made in the PFC. For details, refer to section 21, 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 40 MHz to 100 MHz (160 MHz for the SH7239A and SH7237A).

Maximum operating frequencies differ depending on the products:

- SH7239A and SH7237A:
  Iϕ = 160 MHz, Bϕ = 40 MHz, Pϕ = 40 MHz, Aϕ = 40 MHz, Mϕ = 80 MHz
- SH7239B and SH7237B:
  Iϕ = 100 MHz, Bϕ = 50 MHz, Pϕ = 50 MHz, Aϕ = 50 MHz, Mϕ = 100 MHz

Table 4.3 shows an example of a range for the frequency division ratios that can be specified with FRQCR. Table 4.4 indicates limits on the settings and gives notes on changing the frequencies for the various clock signals.

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
	1/2	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	10 (SH7239A and SH7237A)	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 (SH7239B and SH7237B)	50	25	25	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

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 160 MHz or less, or 100 MHz or less.
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 the maximum operating frequency.
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 40 MHz) or must be lower than 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 40 MHz) or the bus clock (Bϕ) frequency.
7. When using the MTU2S, the MTU clock (Mϕ) frequency must not exceed 100 MHz (or 80 MHz) and exceed the Pϕ and Bϕ frequencies. The MTU 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.

Table 4.4 Limits on Settings and Notes on Changing the Frequency of the Various Clocks

Clock	Limits on Settings for the Clocks: Maximum Value SH7239A, SH7237A	Maximum Value SH7239B, SH7237B	Minimum Value	Selectable Frequency Ratios	Notes on Changes to the Frequency
Internal clock (Iϕ)	160 MHz	100 MHz	40 MHz	1, 1/2, 1/4	Settings must satisfy the following condition. • The frequency is lower than the maximum value.
Bus clock (Bϕ)	40 MHz	50 MHz	20 MHz	1, 1/2, 1/4, 1/8	Settings must satisfy both of the following conditions. • The frequency is lower than the maximum value. • The frequency is lower than that of Iϕ.
Peripheral clock (Pϕ)	40 MHz	50 MHz	20 MHz	1, 1/2, 1/4, 1/8	Settings must satisfy both of the following conditions. • The frequency is lower than the maximum value. • The frequency is lower than that of Bϕ.
AD clock (Aϕ)	40 MHz	50 MHz	40 MHz	1, 1/2, 1/4	Settings must satisfy both of the following conditions. • The frequency is lower than the maximum value. • The frequency is lower than that of Iϕ. • The frequency is higher than that of Pϕ. • The frequency is an integer multiple of that of Pϕ.
MTU clock (Mϕ)	80 MHz	100 MHz	40 MHz	1, 1/2, 1/4	

Note: Other important notes on usage apply to changes to the frequency of the bus clock (Bϕ). For details, refer to section 4.5, Changing the Frequency.

4.4 Register Descriptions

The clock pulse generator has the following registers.

Table 4.5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'0535	H'FFFE0010	16
MTU 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 whether a clock is output from the CK pin in software standby mode, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock (Iϕ) and peripheral clock (Pϕ). FRQCR can be accessed only in word units. Execute the NOP instruction for 32Pϕ clock after having confirmed the set value by reading the 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. Settings must be such that the output clock signal is at a frequency no higher than 40 MHz and no higher than that of Iφ in the case of the SH7239A and SH7237A, and no higher than 50 MHz and no higher than that of Iφ in the case of the SH7239B and SH7237B. 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. Settings must be such that the output clock signal is at a frequency no higher than 160 MHz in the case of the SH7239A and SH7237A and no higher than 100 MHz in the case of the SH7239B and SH7237B. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: Setting prohibited Others: Setting prohibited

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 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. Settings must be such that the output clock signal is at a frequency no higher than 40 MHz and no higher than that of Bϕ in the case of the SH7239A and SH7237A, and no higher than 50 MHz and no higher than that of Bϕ in the case of the SH7239B and SH7237B. 000: × 1 001: × 1/2 010: Setting prohibited 011: × 1/4 100: Setting prohibited 101: × 1/8 Others: Setting prohibited

4.4.2 MTU 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	R	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 frequency of the MTU clock is no higher than 80 MHz in the case of the SH7239A and SH7237A and no higher than 100 MHz in the case of the SH7239B and SH7237B, and is an integer multiple of the peripheral clock frequency (Pϕ). 00: × 1 01: × 1/2 10: Setting prohibited 11: × 1/4

Note: An MTU clock should be set in the range of the internal clock (Iϕ) ≥ the MTU clock (Mϕ).

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	R	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 frequency of the AD clock is no higher than 40 MHz in the case of the SH7239A and SH7237A and no higher than 50 MHz in the case of the SH7239B and SH7237B, and is an integer multiple of the peripheral clock frequency (Pϕ). 00: × 1 01: × 1/2 10: Setting prohibited 11: × 1/4

Note: An AD clock (Aϕ) should be set in the range of the internal clock (Iϕ) ≥ the AD clock (Aϕ).

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	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, MTU clock, and AD clock under the software control through the frequency control register (FRQCR), MTU clock frequency control register (MCLKCR), 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 MTU clock, specify the frequencies to satisfy the following condition: 80 MHz (SH7239A, SH7237A) or 100 MHz (SH7239B, SH7237B) ≥ MTU 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.

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 4.6. 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.

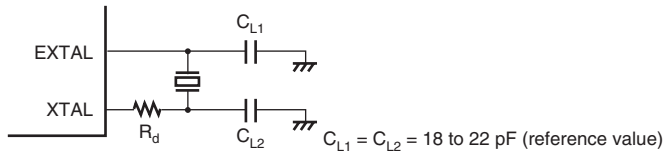


Figure 4.2 Example of Crystal Resonator Connection

Table 4.6 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 4.7.

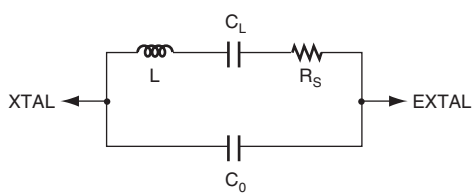


Figure 4.3 Crystal Resonator Equivalent Circuit

Table 4.7 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.

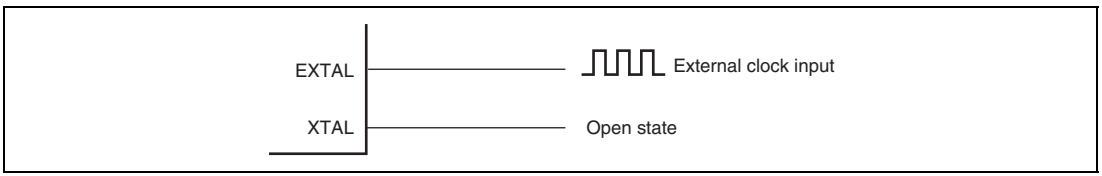


Figure 4.4 Example of External Clock Connection

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. For details, refer to appendix A, Pin States.

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 Notes on Board Design

4.8.1 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.

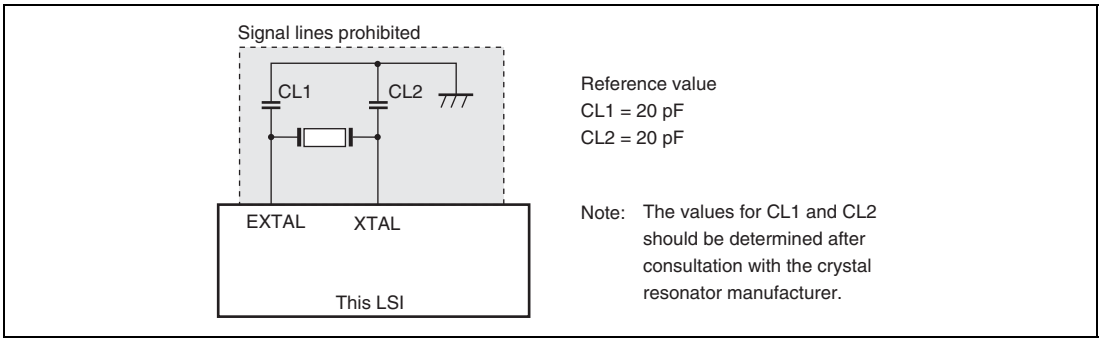
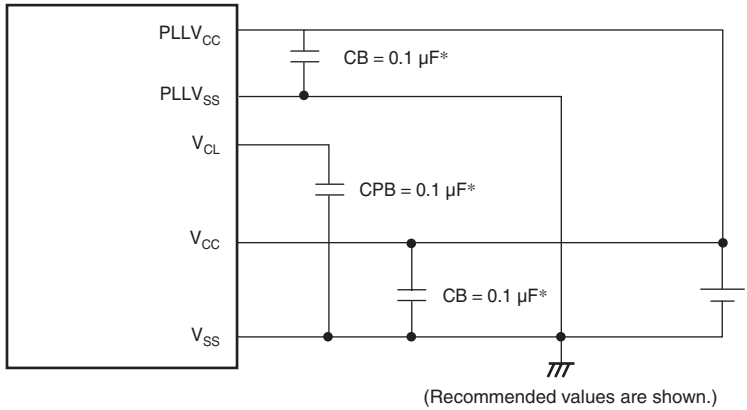


Figure 4.5 Note on Using a Crystal Resonator

A circuitry shown in figure 4.6 is recommended as an external circuitry around the PLL. PLLVCC, PLLVSS, VCL, and VSS 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 VCL and VSS 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 4.6 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, DTC address error		
Instruction	FPU exception*4		
	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)	
		Watchdog timer (WDT)	
		Multi-function timer pulse unit 2 (MTU2)	Low

Type	Exception Handling		Priority
Interrupt	On-chip peripheral modules	Port output enable 2 (POE2): OEI1 and OEI2 interrupts	High
		Multi-function timer pulse unit 2S (MTU2S)	
		Port output enable 2 (POE2): OEI3 interrupt	
		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*4, MOV.L@disp12*4, MOVI20, MOVI20S, MOVU.B, MOVU.W.

4. These instructions are only provided by the SH7239 Group. The operation cannot be guaranteed in the SH7237 Group.

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.

Note: * These instructions are only provided by the SH7239 Group.

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*[1]		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*2	64	H'00000100 to H'00000103
	:	:
	511	H'000007FC to H'000007FF

Notes: 1. Only provided by the SH7239 Group.
2. 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*1**	**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*2	Not initialized
	High	—	Manual reset	Initialized	Not initialized*2	Not initialized

Note: 1. Only provided by the SH7239 Group.
2. 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 (for details, refer to section 27, User Debugging Interface (H-UDI).) 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*	None (normal)
		Instruction fetched from on-chip peripheral module space*	Address error occurs
		Instruction fetched from external memory space in single-chip mode	Address error occurs
Data read/write	CPU, DMAC, or DTC	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) (SH7239A and SH7237A only), 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 IRQ6 pins (external input)	7
Memory error	Flash memory (ROM), data flash (FLD)	1
On-chip peripheral module	A/D converter (ADC)	3
	Controller area network (RCAN-ET)	4
	Direct memory access controller (DMAC)	16
	Compare match timer (CMT)	2
	Watchdog timer (WDT)	1
	Multi-function timer pulse unit 2 (MTU2)	28
	Multi-function timer pulse unit 2S (MTU2S)	13
	Port output enable 2 (POE2)	3
	Renesas serial peripheral interface (RSPI)	3
	Serial communication interface (SCI)	12
	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 18 (IPR01, IPR02, and IPR05 to IPR18) 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 18 (IPR01, IPR02, IPR05 to IPR18), for details of IPR01, IPR02, and IPR05 to IPR18.

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

Note: * These instructions are only provided by the SH7239 Group. The operation cannot be guaranteed in the SH7237 Group.

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 (SH7239 Group Only)

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*2**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*1	Not accepted	Not accepted	Not accepted	Not accepted

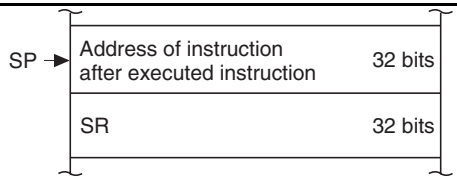
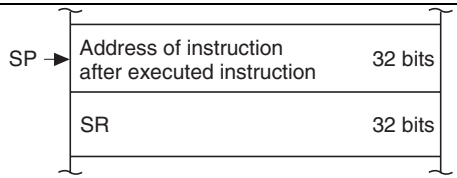
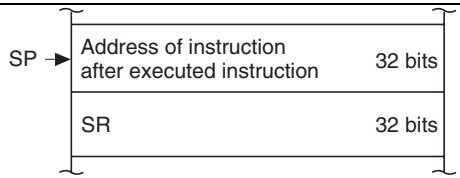
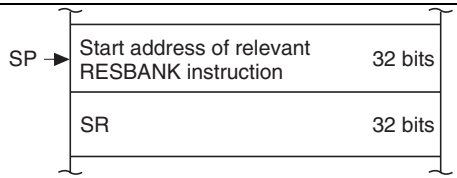
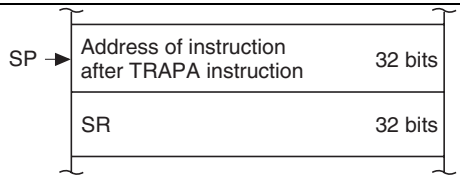
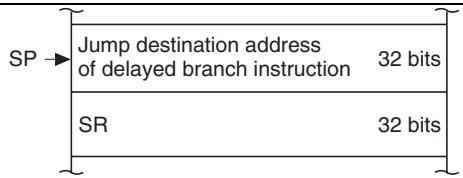
Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

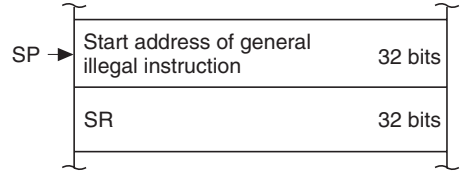
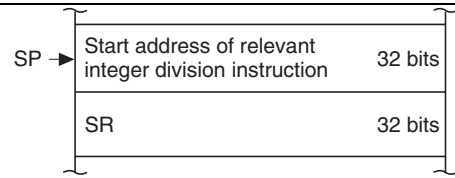
2. Only provided by the SH7239 Group. The operation cannot be guaranteed in the SH7237 Group.

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)	

Note: * Only provided by the SH7239 Group. The operation cannot be guaranteed in the SH7237 Group.

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.

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 16 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.

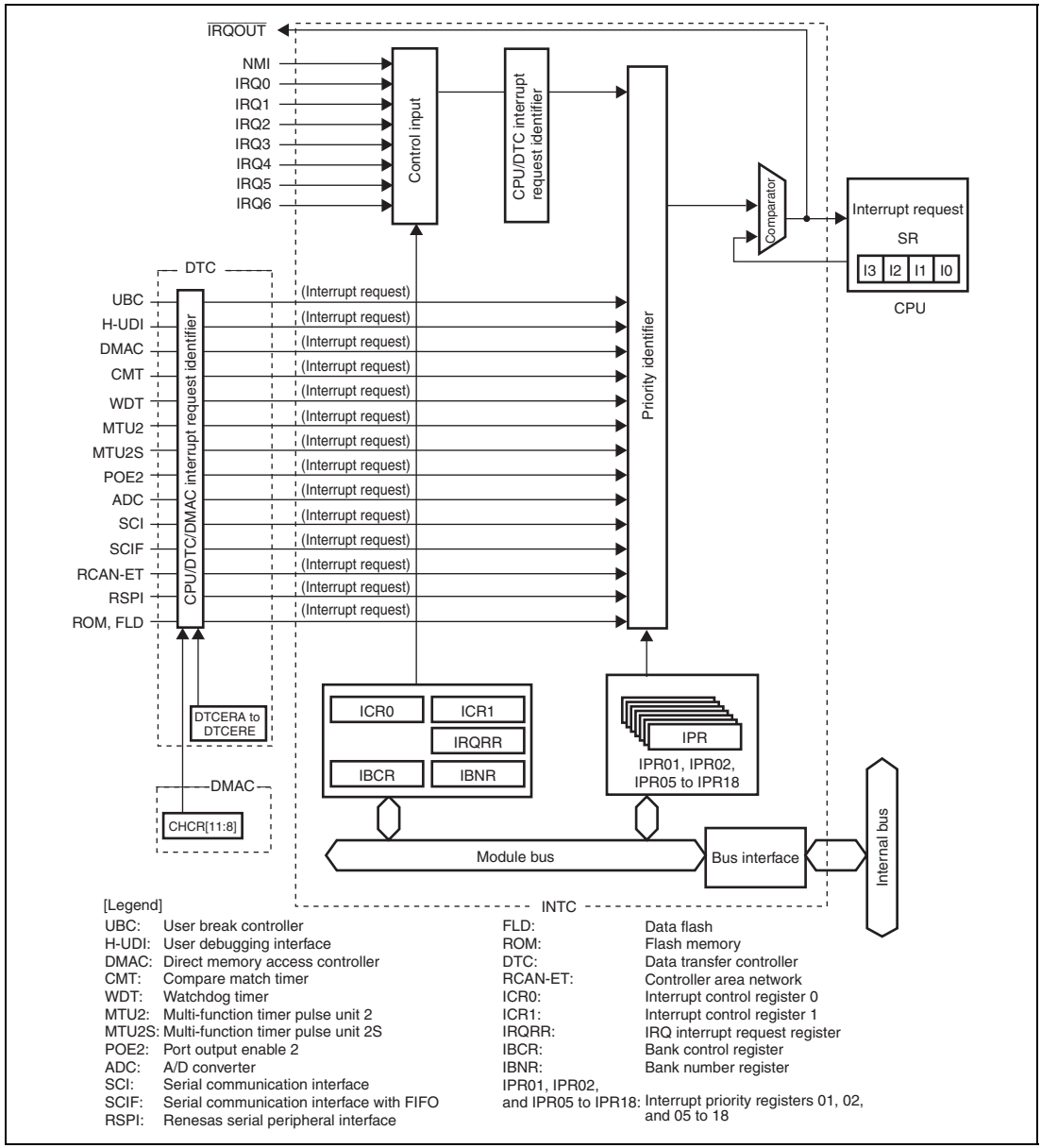


Figure 6.1 Block Diagram of INTC

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	IRQ6 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

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 18 (IPR01, IPR02, IPR05 to IPR18)

IPR01, IPR02, and IPR05 to IPR18 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 IPR18.

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 IPR18

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	Reserved
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	Reserved	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)	Reserved	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	Reserved	ADI2	Reserved
Interrupt priority register 18	Reserved	RCAN-ET	Reserved	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 IPR18 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 IRQ6 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
	-	-	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	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

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	IRQ61S	0	R/W	IRQ Sense Select These bits select whether interrupt signals corresponding to pins IRQ6 to IRQ0 are detected by a low level, falling edge, rising edge, or both edges. 00: Interrupt request is detected on low level of IRQn input 01: Interrupt request is detected on falling edge of IRQn input 10: Interrupt request is detected on rising edge of IRQn input 11: Interrupt request is detected on both edges of IRQn input
12	IRQ60S	0	R/W	
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
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 = 6 to 0

6.3.4 IRQ Interrupt Request Register (IRQRR)

IRQRR is a 16-bit register that indicates interrupt requests from external input pins IRQ6 to IRQ0. If edge detection is set for the IRQ6 to IRQ0 interrupts, writing 0 to the IRQ6F to IRQ0F bits after reading IRQ6F 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
	-	-	-	-	-	-	-	-	-	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	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 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	IRQ6F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ6 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred 1: IRQn interrupt has occurred [Clearing condition] • IRQn input is high [Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected 1: IRQn interrupt request is 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. [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin
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)*	
0	IRQ0F	0	R/(W)*	

[Legend]

n = 6 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 BN[3:0] bits.

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.

6.4 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 27, User Debugging Interface (H-UDI).

6.4.4 IRQ Interrupts

IRQ interrupts are input from pins IRQ6 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 (IRQ61S to IRQ01S and IRQ60S 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 IRQ6 to IRQ0 pins are low. An interrupt request signal is stopped being sent to the INTC when the IRQ6 to IRQ0 pins are driven high. The status of the interrupt requests can be checked by reading the IRQ interrupt request bits (IRQ6F 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 IRQ6 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 IRQ6F 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 23, 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)
- Watchdog timer (WDT)
- Multi-function timer pulse unit 2 (MTU2)
- Multi-function timer pulse unit 2S (MTU2S)
- Port output enable 2 (POE2)
- 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 18 (IPR05 to IPR18). 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 18 (IPR01, IPR02, and IPR05 to IPR18). However, if two or more interrupts specified by the same IPR among IPR05 to IPR18 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)	—	
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)	—	
	ADI2	100	H'00000190 to H'00000193	0 to 15 (0)	IPR17 (7 to 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
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)	—	
WDT	ITI		152	H'00000260 to H'00000263	0 to 15 (0)	IPR08 (3 to 0)	—	
MTU2	MTU2_0	TGIA_0	156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (15 to 12)	1	
		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	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_2	TGIA_2	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR10 (15 to 12)	1	High
		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	
	MTU2_3	TGIA_3	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR10 (7 to 4)	1	
		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	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
POE2	OEI1		200	H'00000320 to H'00000323	0 to 15 (0)	IPR11 (3 to 0)	1	High
	OEI2		201	H'00000324 to H'00000327			2	
MTU2S	MTU2S_3	TGIA_3S	204	H'00000330 to H'00000333	0 to 15 (0)	IPR12 (15 to 12)	1	
		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)	—	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
RSPI	SPEI		233	H'000003A4 to H'000003A7	0 to 15 (0)	IPR17 (15 to 12)	1	High
	SPRI		234	H'000003A8 to H'000003AB			2	
	SPTI		235	H'000003AC to H'000003AF			3	
SCI	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	
	SCI1	ERI1	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR16 (11 to 8)	1	
		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	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
SCIF	SCIF3	BRI3	252	H'000003F0 to H'000003F3	0 to 15 (0)	IPR14 (3 to 0)	1	High
		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 18 (IPR01, IPR02, and IPR05 to IPR18). 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.

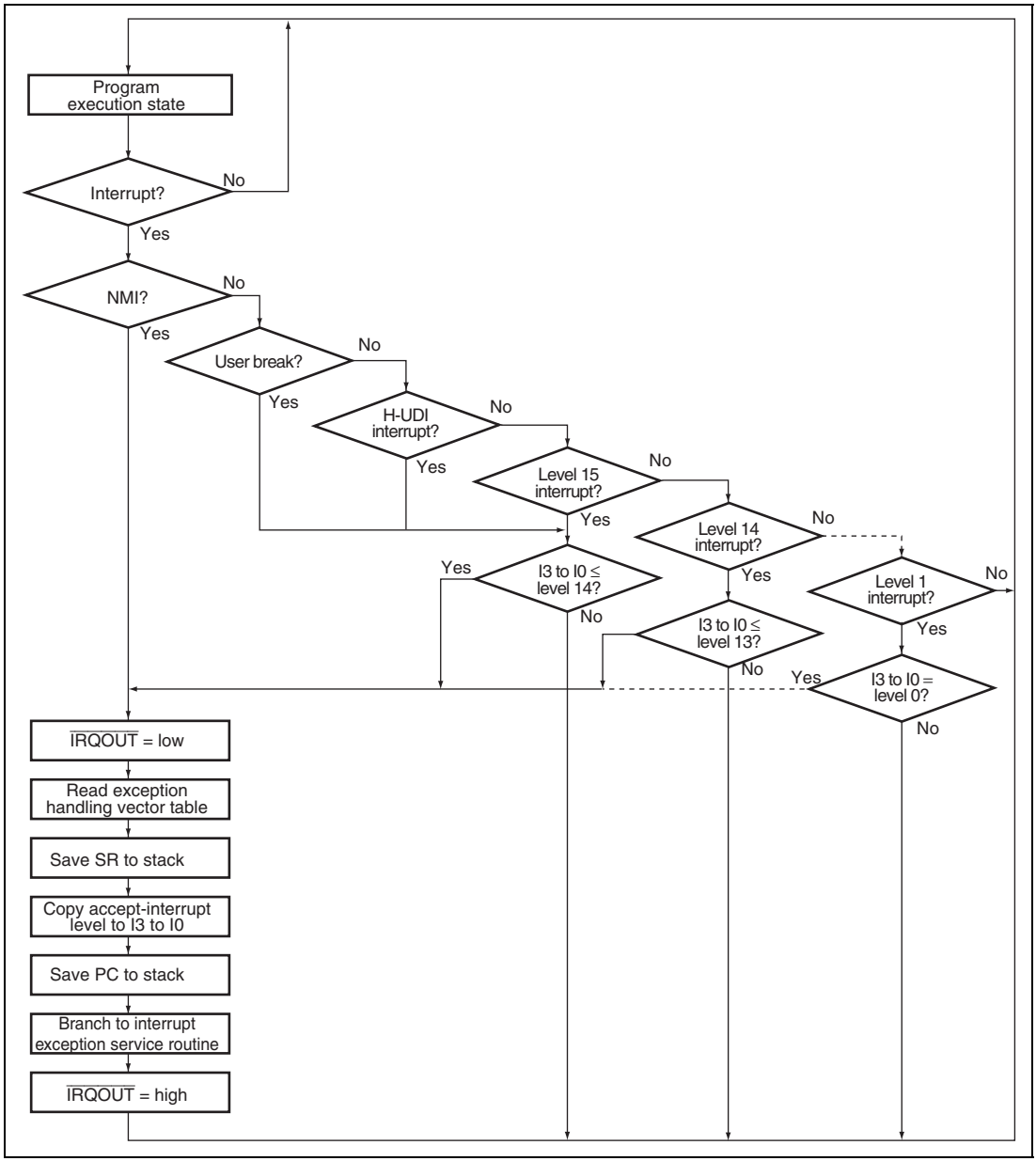
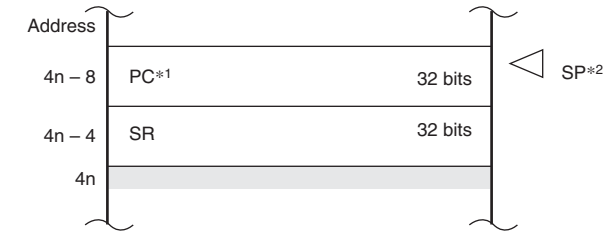


Figure 6.2 Interrupt Operation Flow

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: NMI	UBC	H-UDI	IRQ	Peripheral Module	Remarks
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 160-MHz operation*1*3: 0.050 to 0.144 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 160-MHz operation*1*3: 0.075 to 0.169 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 160-MHz operation*1*3: 0.069 to 0.144 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 160-MHz operation*1*3: 0.125 to 0.200 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 160-MHz operation*1*3: 0.069 to 0.144 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 160-MHz operation*1*3: 0.188 to 0.263 µs

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

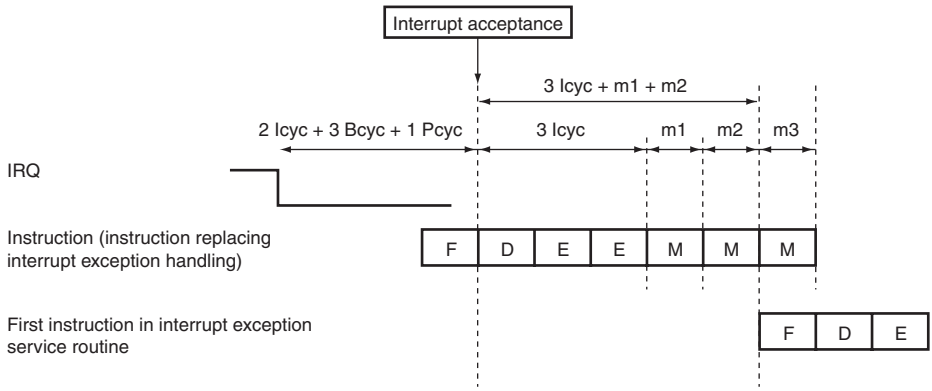
m1: Vector address read (longword read)

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) = (160 MHz, 40 MHz, 40 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)

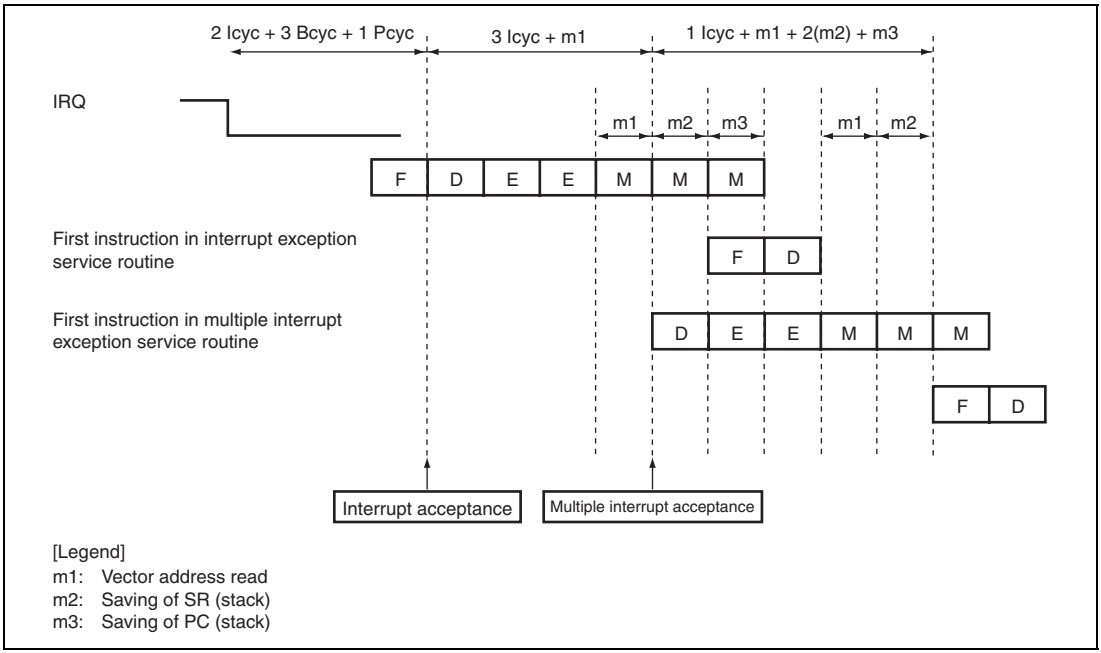


Figure 6.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

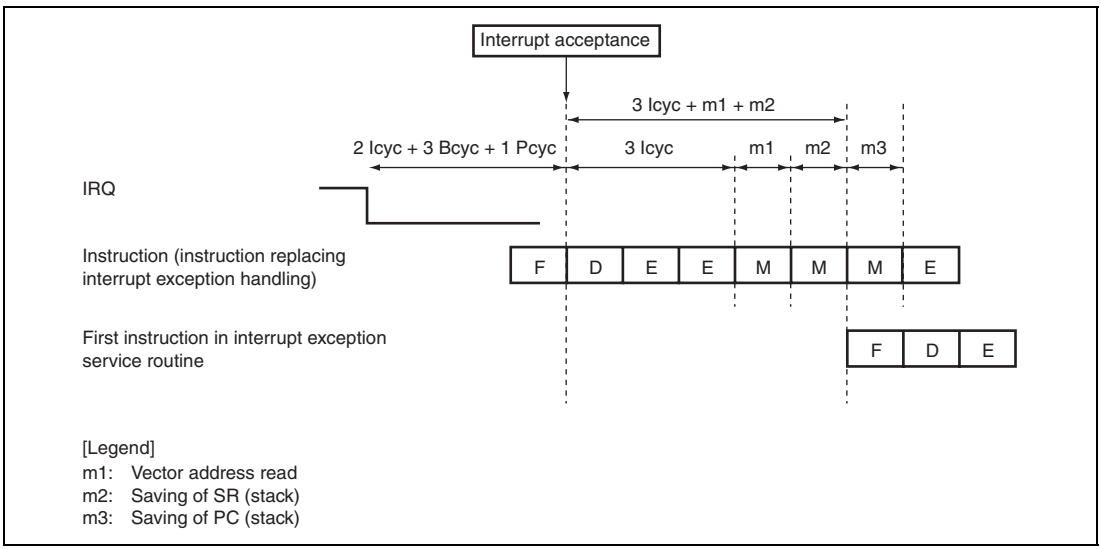


Figure 6.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

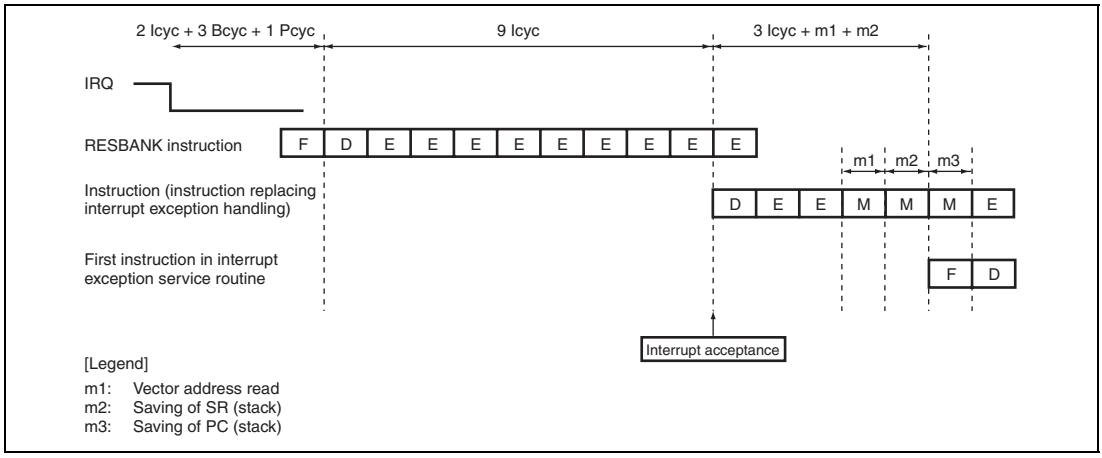


Figure 6.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

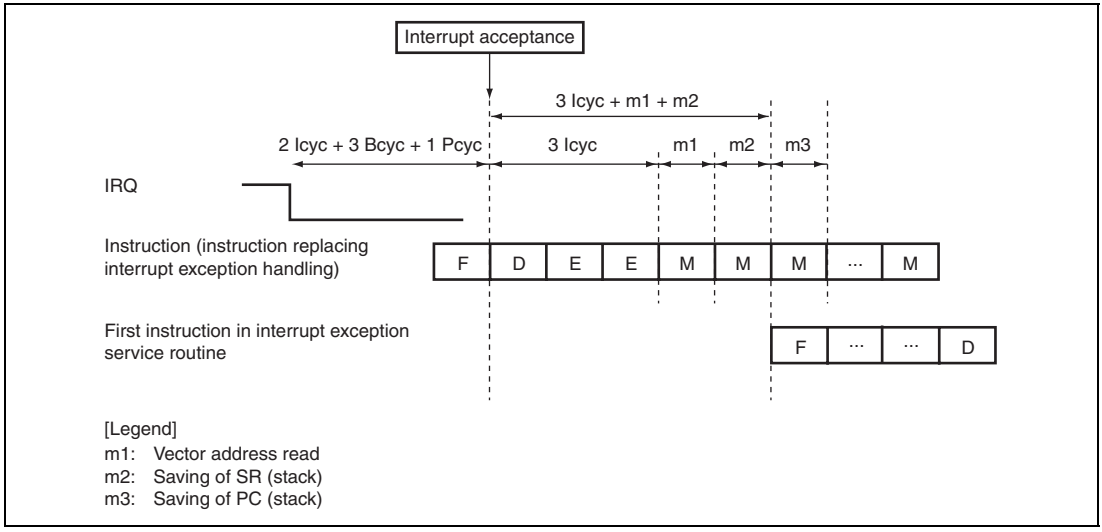


Figure 6.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

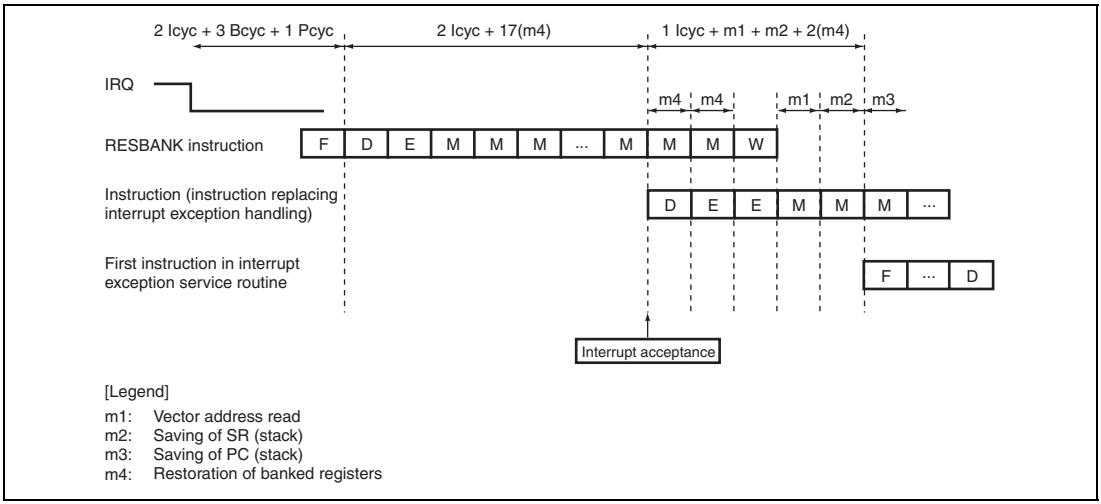


Figure 6.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

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.

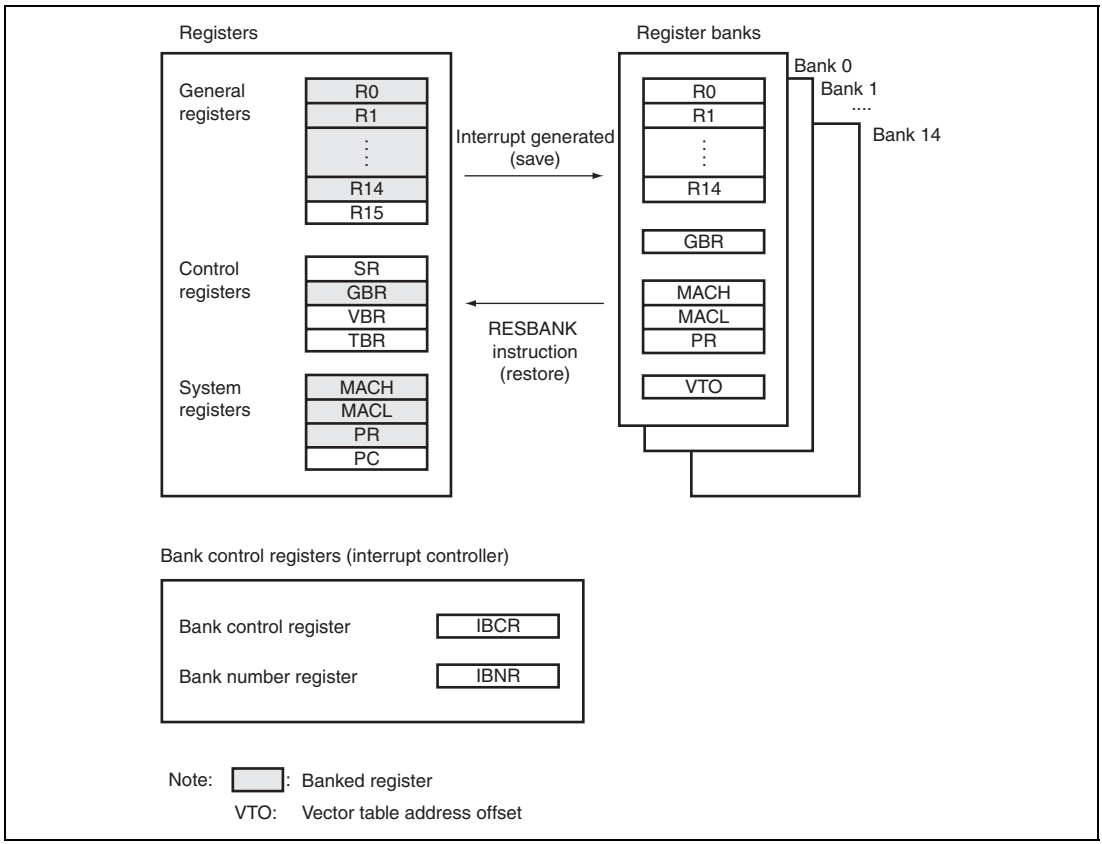


Figure 6.10 Overview of Register Bank Configuration

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.

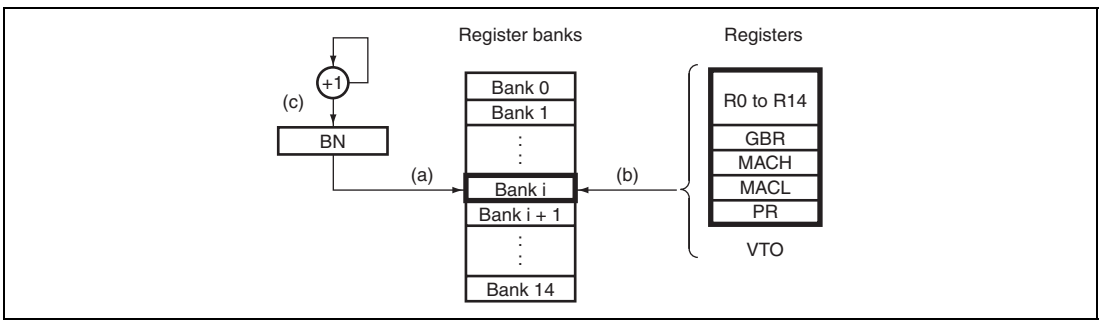


Figure 6.11 Bank Save Operations

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.

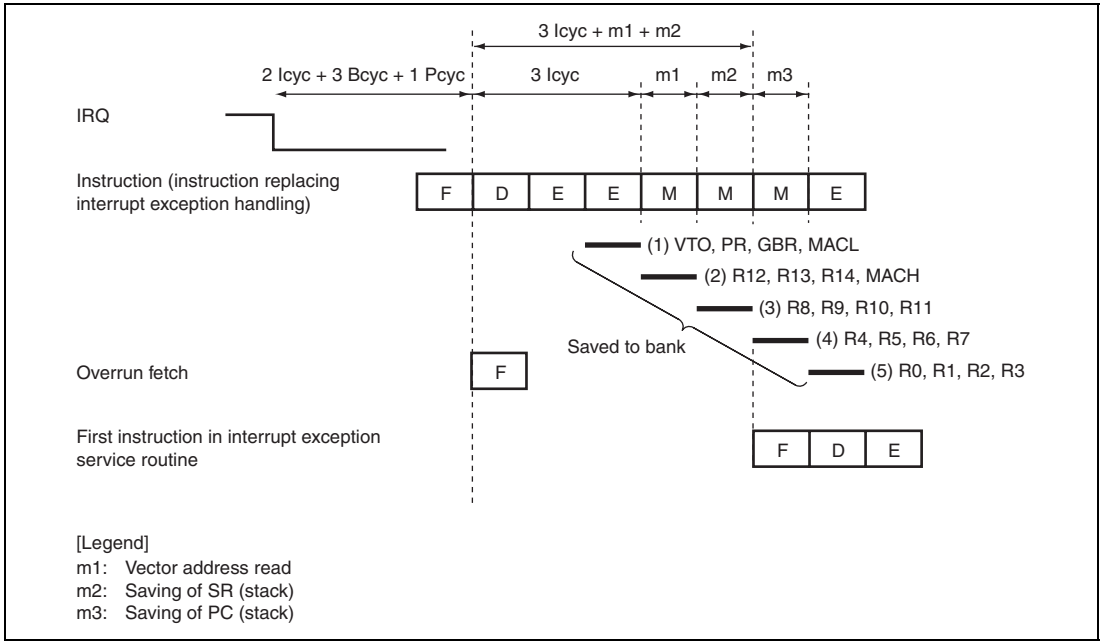


Figure 6.12 Bank Save Timing

(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 Interrupt Requests

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.

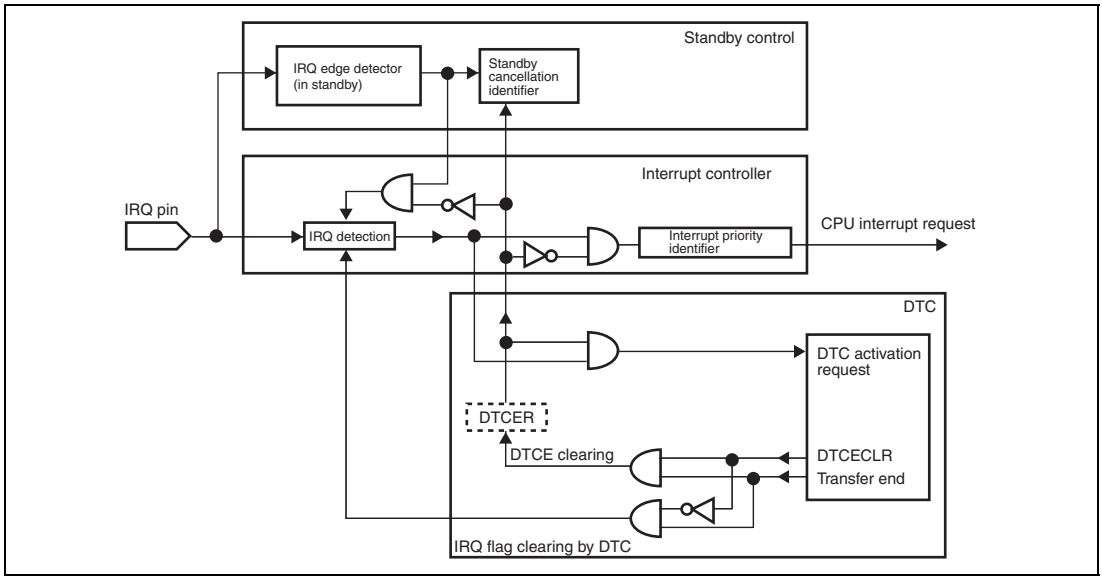


Figure 6.13 Interrupt Control Block Diagram

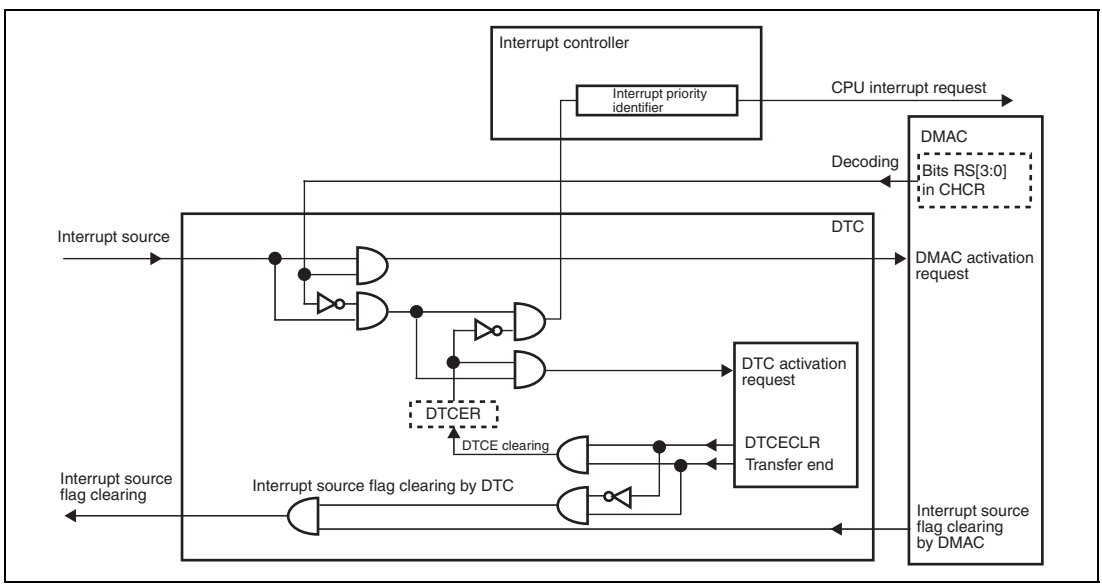


Figure 6.14 Block Diagram of Controlling an On-Chip Peripheral Module Interrupt

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 Note

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, confirm that it is 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} via a resistor.

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.

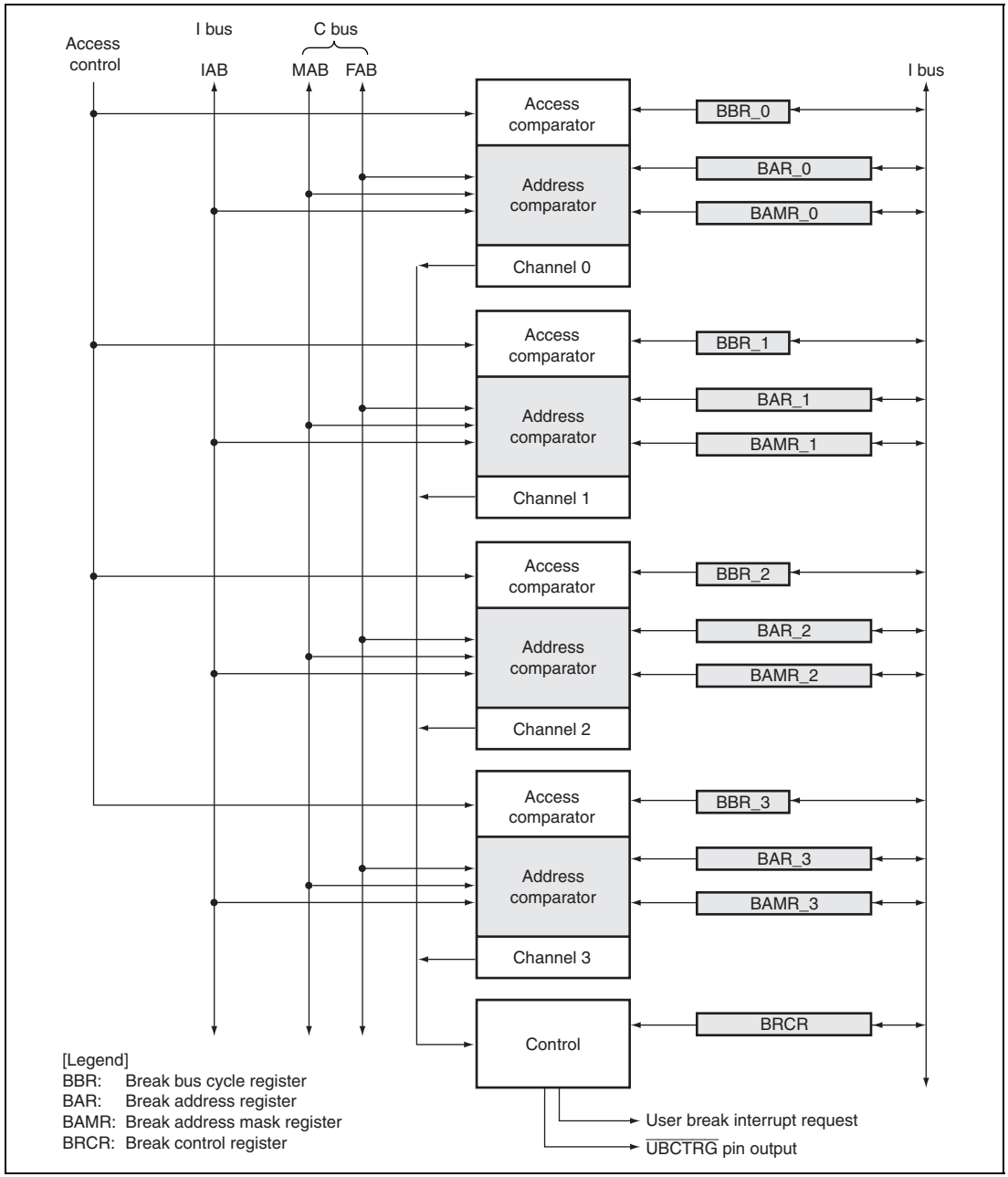


Figure 7.1 Block Diagram of UBC

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 by the CD0[1:0] bits. 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 by the CD1[1:0] bits. 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 by the CD2[1:0] bits. 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 by the CD3[1:0] bits. 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. 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

  This is only selectable after data transfer every time chain transfer is conducted or only after the specified number of data-transfer operations (transfer starts when the counter is zero).
- 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
- This is only selectable after data transfer every time chain transfer is conducted or only after the specified number of data-transfer operations (transfer starts when the counter is zero).

  A CPU interrupt can be requested after one data transfer completion

  A CPU interrupt can be requested after the specified data transfer completion
- The read skip function and write-back skip function for the transfer information can be used to shorten DTC transfer time.
- Module stop mode can be used to reduce power consumption.
- Short address mode can be used to shorten the time for DTC activation.
- 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.

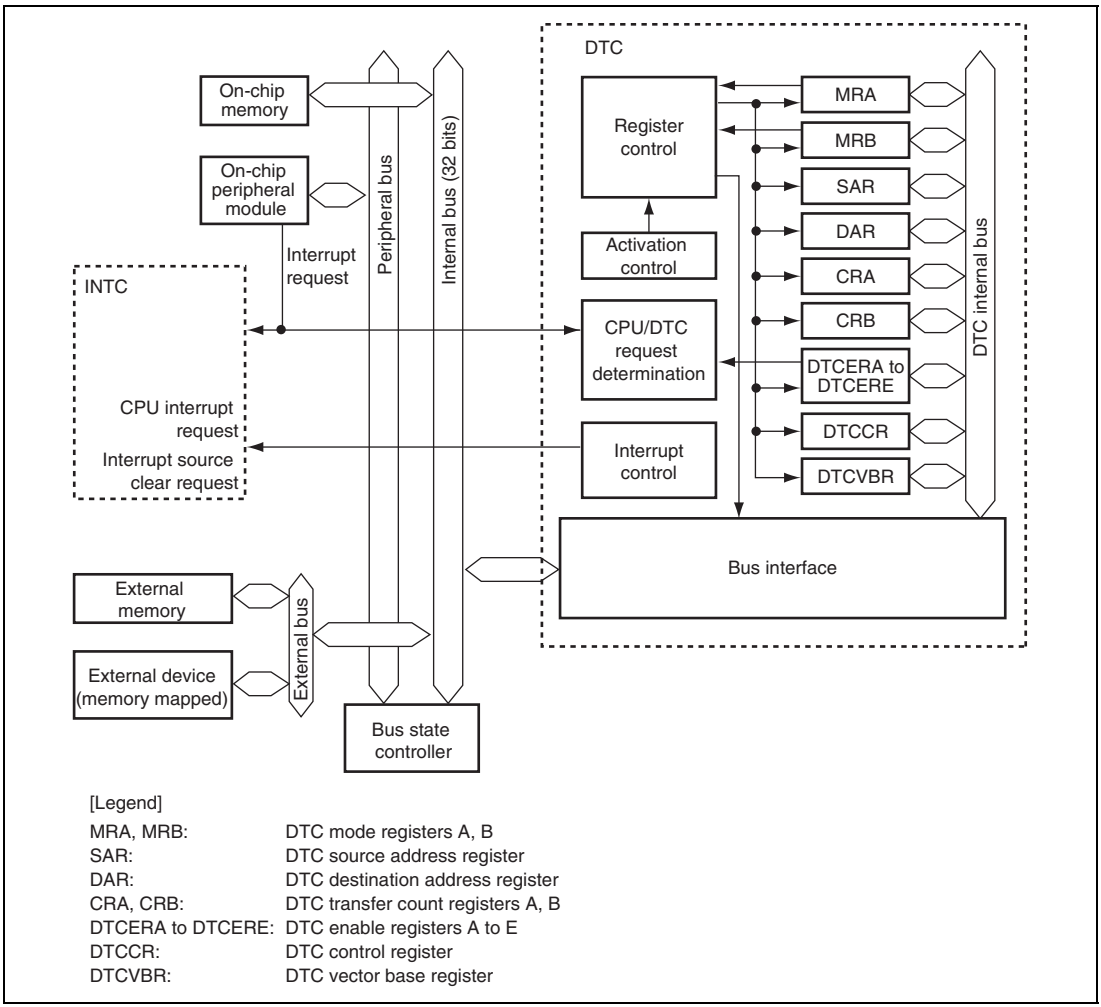


Figure 8.1 Block Diagram of DTC

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 28, 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 (number of data transfers as specified for the transfer counter by CRA or CRB) of data transfers ends.

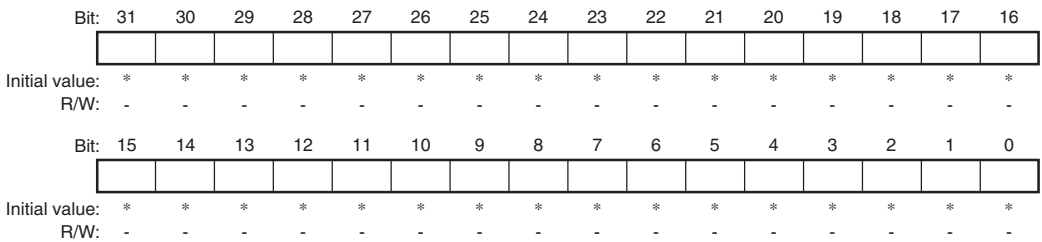
Bit	Bit Name	Initial Value	R/W	Description
4	DTS	Undefined	—	DTC Transfer Mode Select Specifies either the source or destination as repeat or block area during repeat or block transfer mode. 0: Specifies the destination as repeat or block area 1: Specifies the source as repeat or block area
3, 2	DM[1:0]	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
14	DTCE14	0	R/W	Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source.
13	DTCE13	0	R/W	[Clearing conditions]
12	DTCE12	0	R/W	• When writing 0 to the bit to be cleared after reading 1
11	DTCE11	0	R/W	• When the DISEL bit is 1 and the data transfer has ended
10	DTCE10	0	R/W	• When the specified number of transfers have ended
9	DTCE9	0	R/W	These bits are not cleared when the DISEL bit is 0 and the specified number of transfers have not ended
8	DTCE8	0	R/W	[Setting condition]
7	DTCE7	0	R/W	• Writing 1 to the bit after reading 0
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 9.4.4, 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 9.4.4, 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.

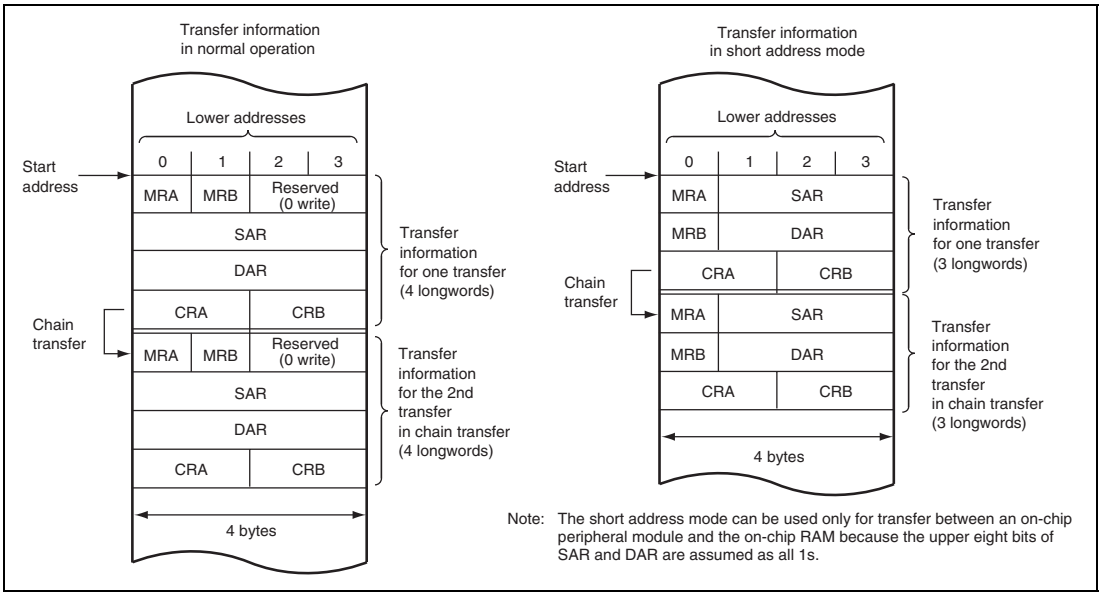


Figure 8.2 Transfer Information on Data Area

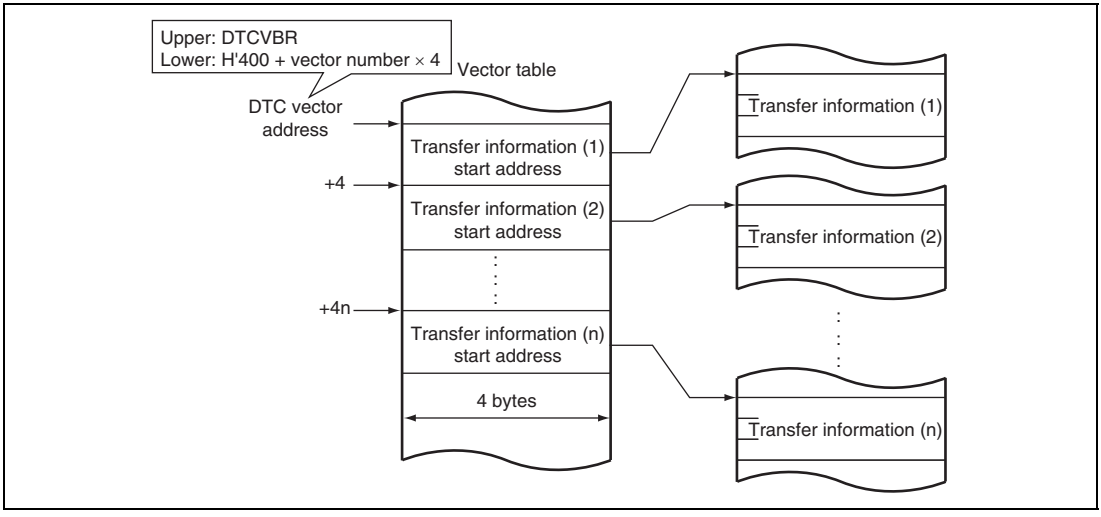


Figure 8.3 Correspondence between DTC Vector Address and Transfer Information

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	
A/D converter	ADI0	92	H'00000570	DTCERA7	ADDR0 to ADDR3	Any location*2	
	ADI1	96	H'00000580	DTCERA6	ADDR4 to ADDR7	Any location*2	
	ADI2	100	H'00000590	DTCERA5	ADDR8 to ADDR15	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	
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	
MTU2_CH2	TGIA_2	172	H'000006B0	DTCERB9	Any location*2	Any location*2	
	TGIB_2	173	H'000006B4	DTCERB8	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_CH3	TGIA_3	180	H'000006D0	DTCERB7	Any location*2	Any location*2	High
	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	
RSPI	SPRI	234	H'000007A8	DTCERD5	SPDR	Any location*2	
	SPTI	235	H'000007AC	DTCERD4	Any location*2	SPDR	
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	Low

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address Offset	DTCE*1	Transfer Source	Transfer Destination	Priority
SCI2	RXI2	249	H'000007E4	DTCERE11	SCRDR_2	Any location*2	High
	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).

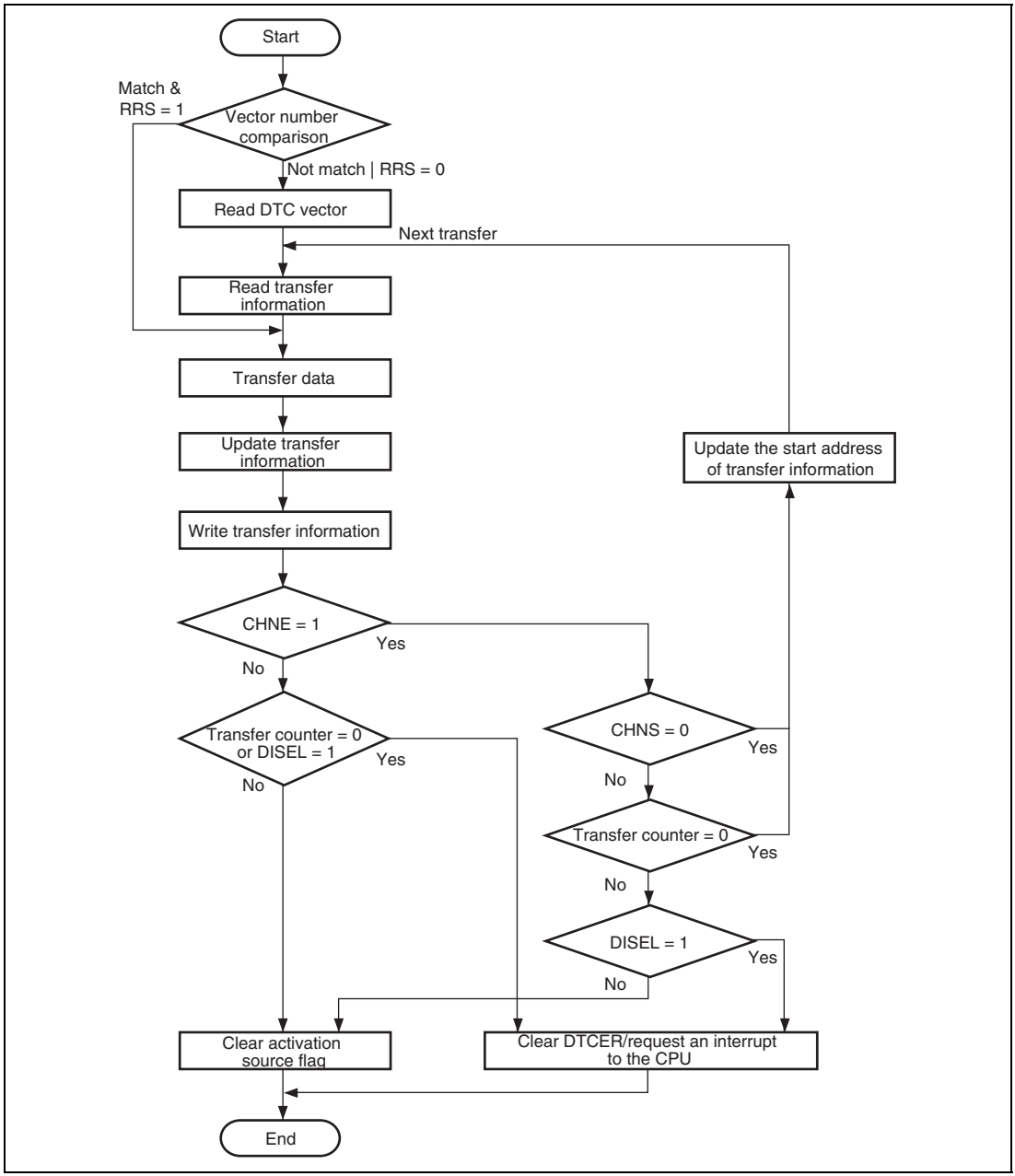


Figure 8.4 Flowchart of DTC Operation

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

	1st Transfer					2nd Transfer					
Transfer Mode	**CHNE**	**CHNS**	**RCHNE**	**DISEL**	**Transfer Counter**[1]	**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.

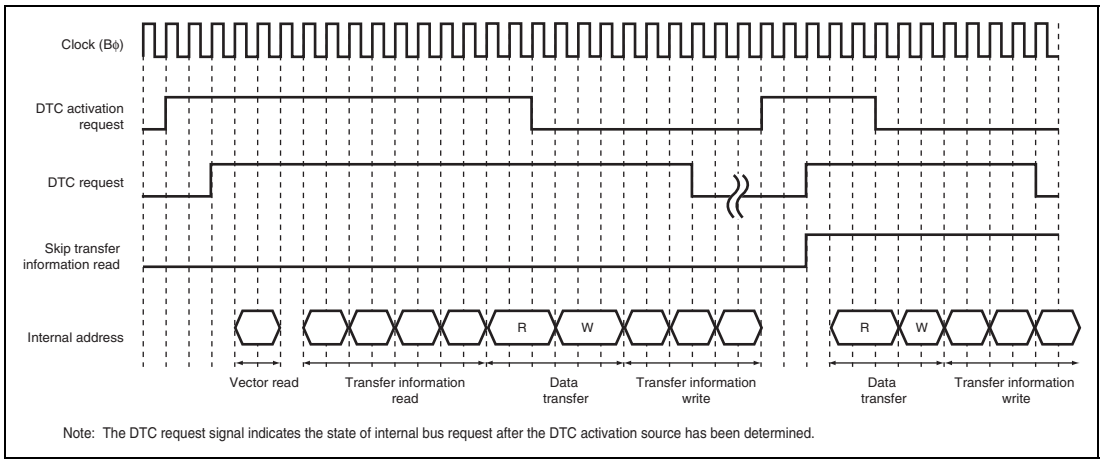


Figure 8.5 Transfer Information Read Skip Timing (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)

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.

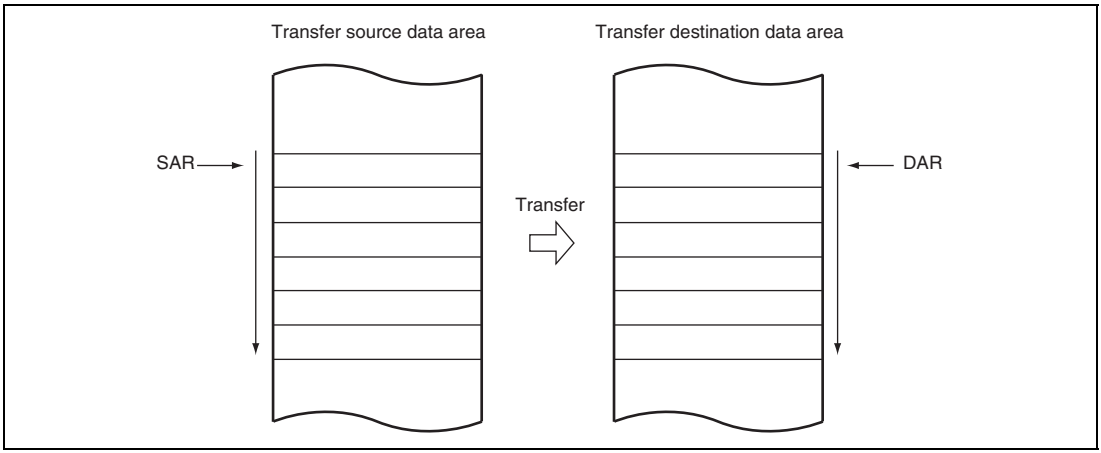


Figure 8.6 Memory Map in Normal Transfer Mode

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.

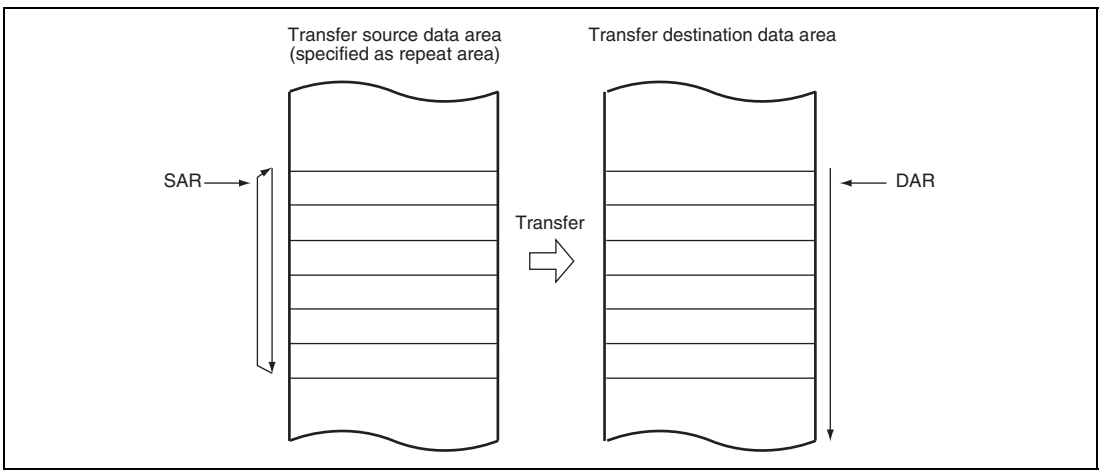


Figure 8.7 Memory Map in Repeat Transfer Mode (When Transfer Source is Specified as Repeat Area)

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.

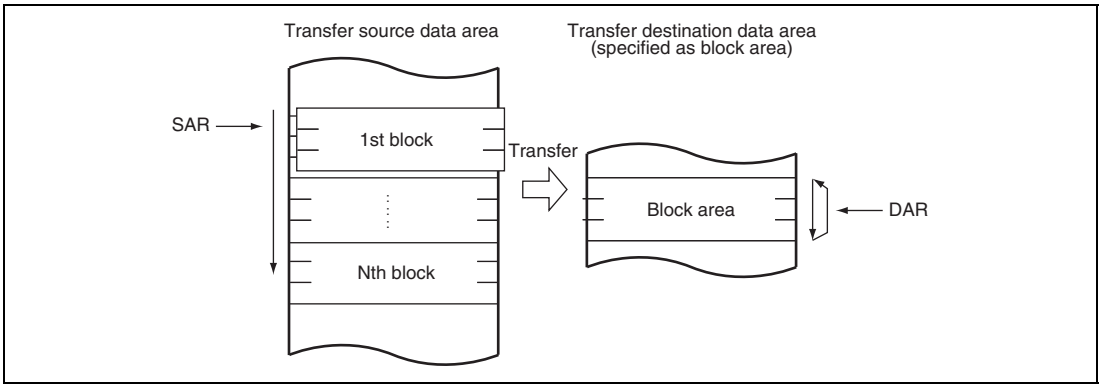


Figure 8.8 Memory Map in Block Transfer Mode (When Transfer Destination is Specified as Block Area)

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.

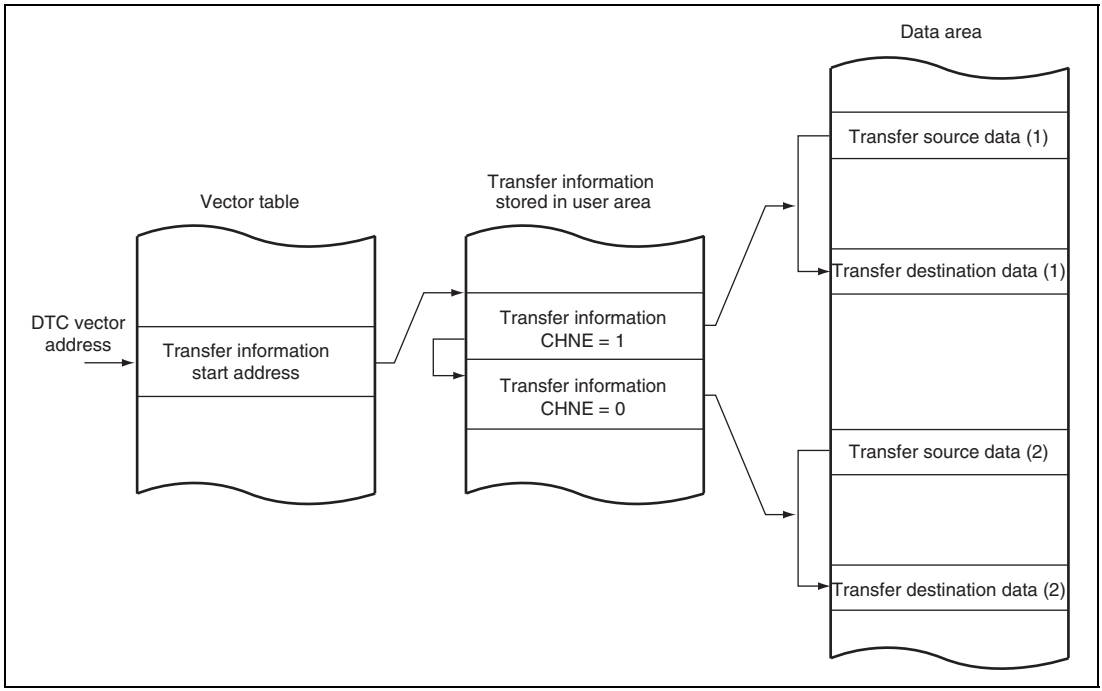
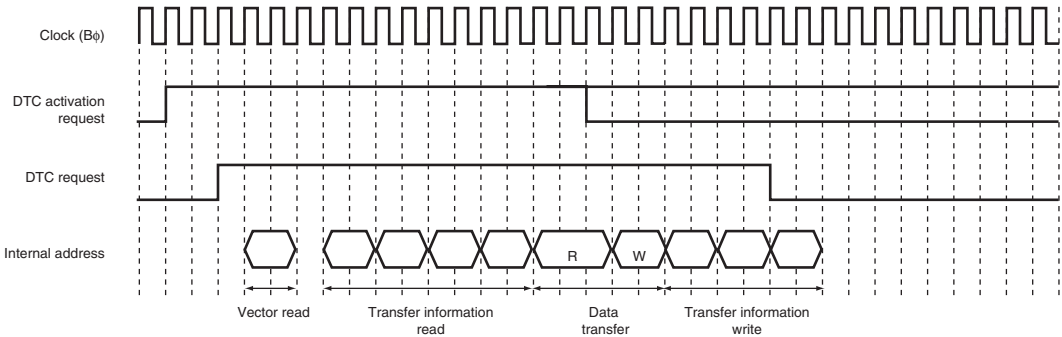


Figure 8.9 Operation of Chain Transfer

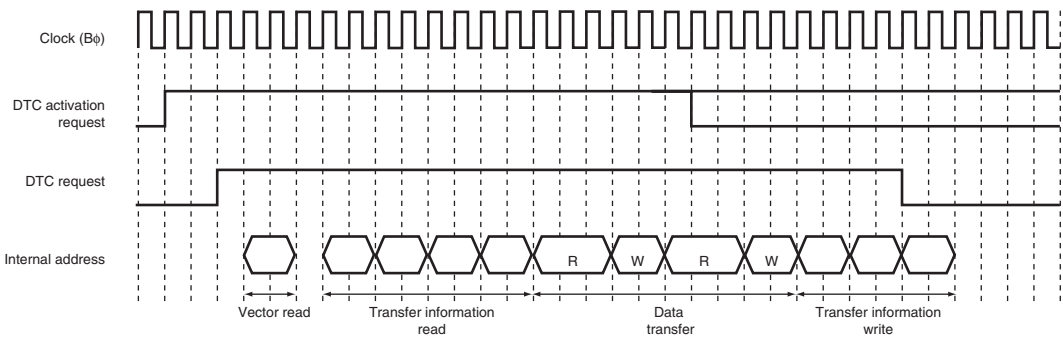
8.5.7 Operation Timing

Figures 8.10 to 8.15 show the DTC operation timings.



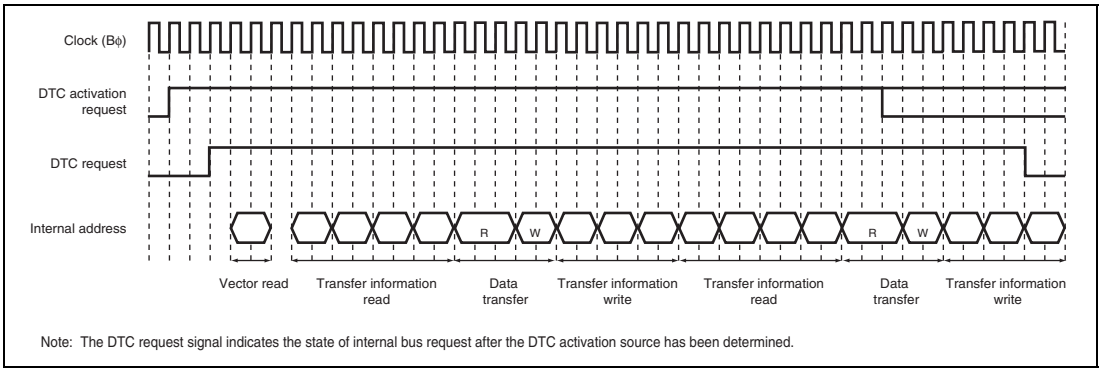
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

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)



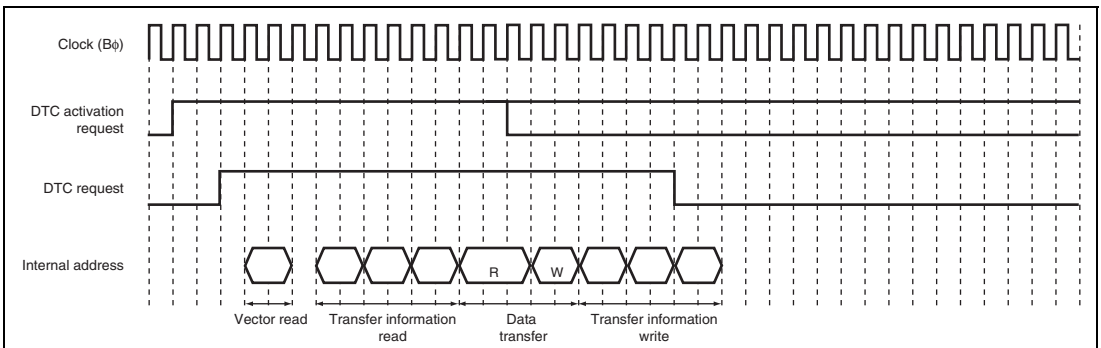
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

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)



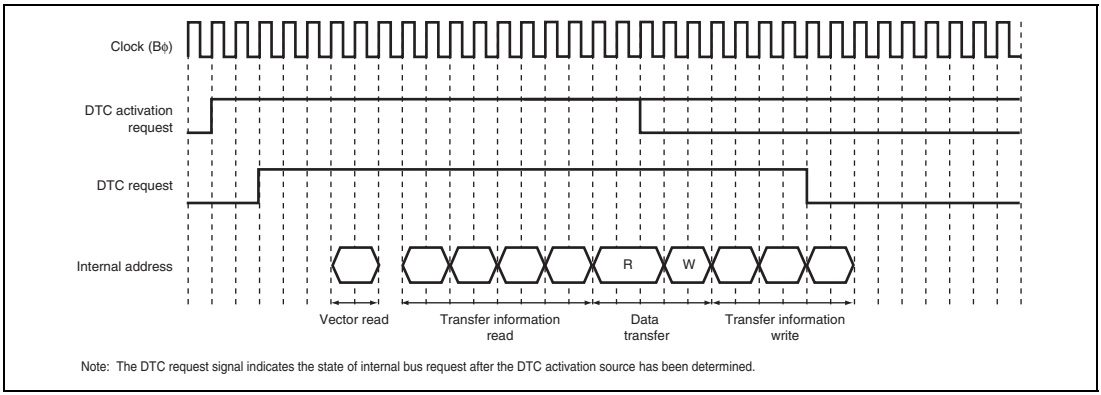
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

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)



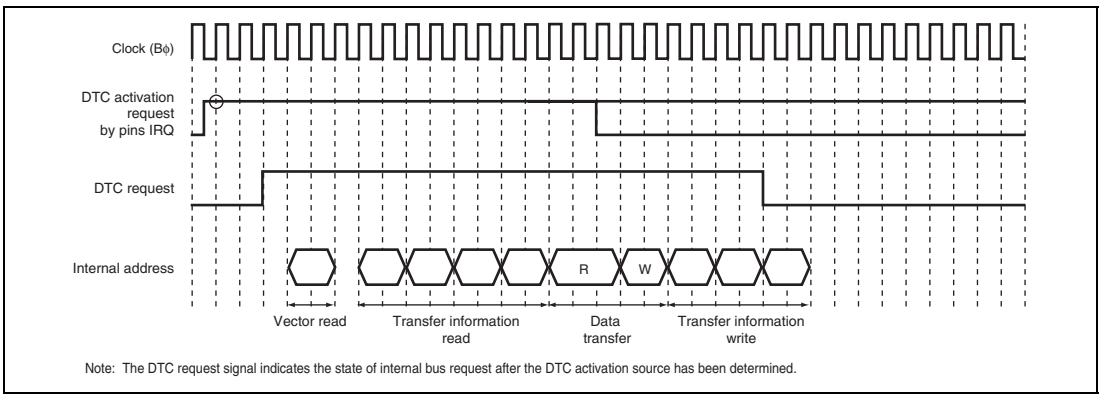
Note: The DTC request signal indicates the state of internal bus request after the DTC activation source has been determined.

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			External Device*4	
Bus width		32 bits	32 bits	8 bits	16 bits	32 bits	8 bits	16 bits
Access cycles		1Bϕ to 4Bϕ*1	3Bϕ to 4Iϕ + 3Bϕ*2	2Pϕ	2Pϕ	2Pϕ	2Bϕ	2Bϕ
Execution status	Vector read S_I	1Bϕ to 4Bϕ*1	3Bϕ to 4Iϕ + 3Bϕ*2	—	—	—	9Bϕ	5Bϕ
	Transfer information read S_J	1Bϕ to 4Bϕ*1	—	—	—	—	9Bϕ	5Bϕ
	Transfer information write S_K	1Bϕ to 3Bϕ*1	—	—	—	—	2Bϕ*5	2Bϕ*5
	Byte data read S_L	1Bϕ to 4Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	3Bϕ	3Bϕ
	Word data read S_L	1Bϕ to 4Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	5Bϕ	3Bϕ
	Longword data read S_L	1Bϕ to 4Bϕ*1	—	1Bϕ + 4Pϕ*3	1Bϕ + 2Pϕ*3	1Bϕ + 4Pϕ*3	9Bϕ	5Bϕ
	Byte data write S_M	1Bϕ to 3Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	2Bϕ*5	2Bϕ*5
	Word data write S_M	1Bϕ to 3Bϕ*1	—	1Bϕ + 2Pϕ*3	1Bϕ + 2Pϕ*3	—	2Bϕ*5	2Bϕ*5
	Longword data write S_M	1Bϕ to 3Bϕ*1	—	1Bϕ + 4Pϕ*3	1Bϕ + 2Pϕ*3	1Bϕ + 4Pϕ*3	2Bϕ*5	2Bϕ*5
	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 to 4Bϕ	2Bϕ to 3Bϕ
Iϕ:Bϕ = 1:1/2	2Bϕ to 3Bϕ	2Bϕ
Iϕ:Bϕ = 1:1/4	2Bϕ	1Bϕ to 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	Write
Iϕ:Bϕ = 1:1	4Iϕ + 3Bϕ	4Iϕ + 3Bϕ
Iϕ:Bϕ = 1:1/2	4Iϕ + 3Bϕ	4Iϕ + 3Bϕ
Iϕ:Bϕ = 1:1/4	4Iϕ + 3Bϕ	4Iϕ + 3Bϕ
Iϕ:Bϕ = 1:1/8	3Bϕ	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. 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.
5. 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 9.5.8 (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 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 9.4.4, 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 : 4 : 4, or 8 : 4 : 4 : 8 : 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 restrictions apply to setting 3.
- The DTPR bit in BSCEHR should be 0. Setting 1 is prohibited.

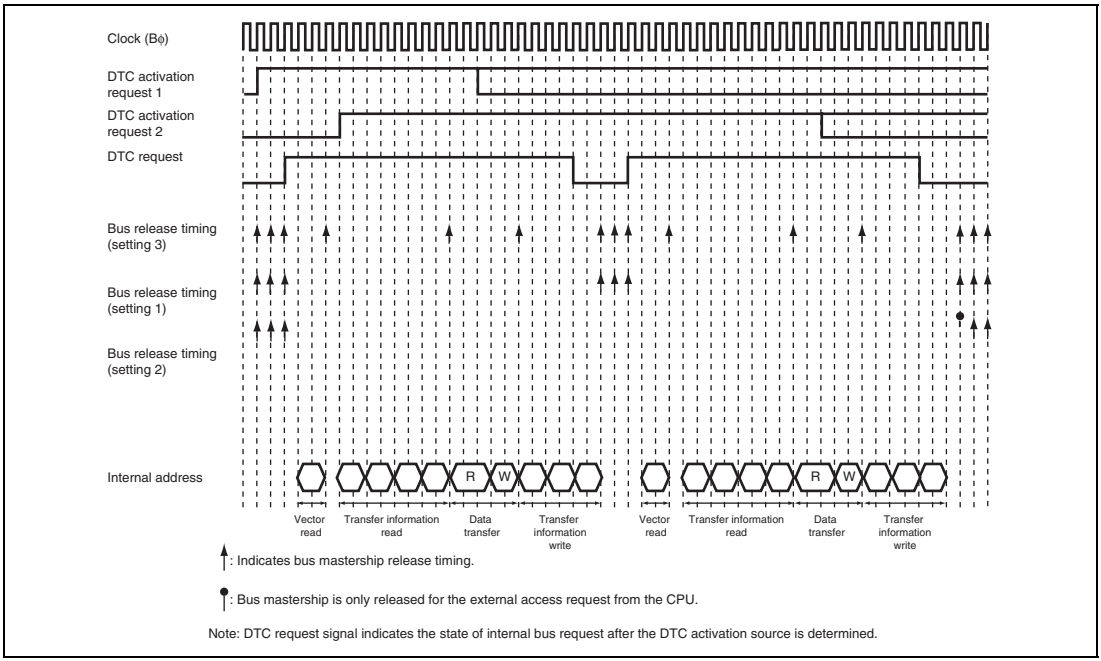


Figure 8.16 Example of DTC Operation Timing: Conflict of Two Activation Requests in Normal Transfer Mode (Activated by On-Chip Peripheral Module; Iϕ : Bϕ : Pϕ = 1 : 1/2 : 1/2; Data Transferred from On-Chip Peripheral Module to On-Chip RAM; Transfer Information is Written in 3 Cycles)

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.

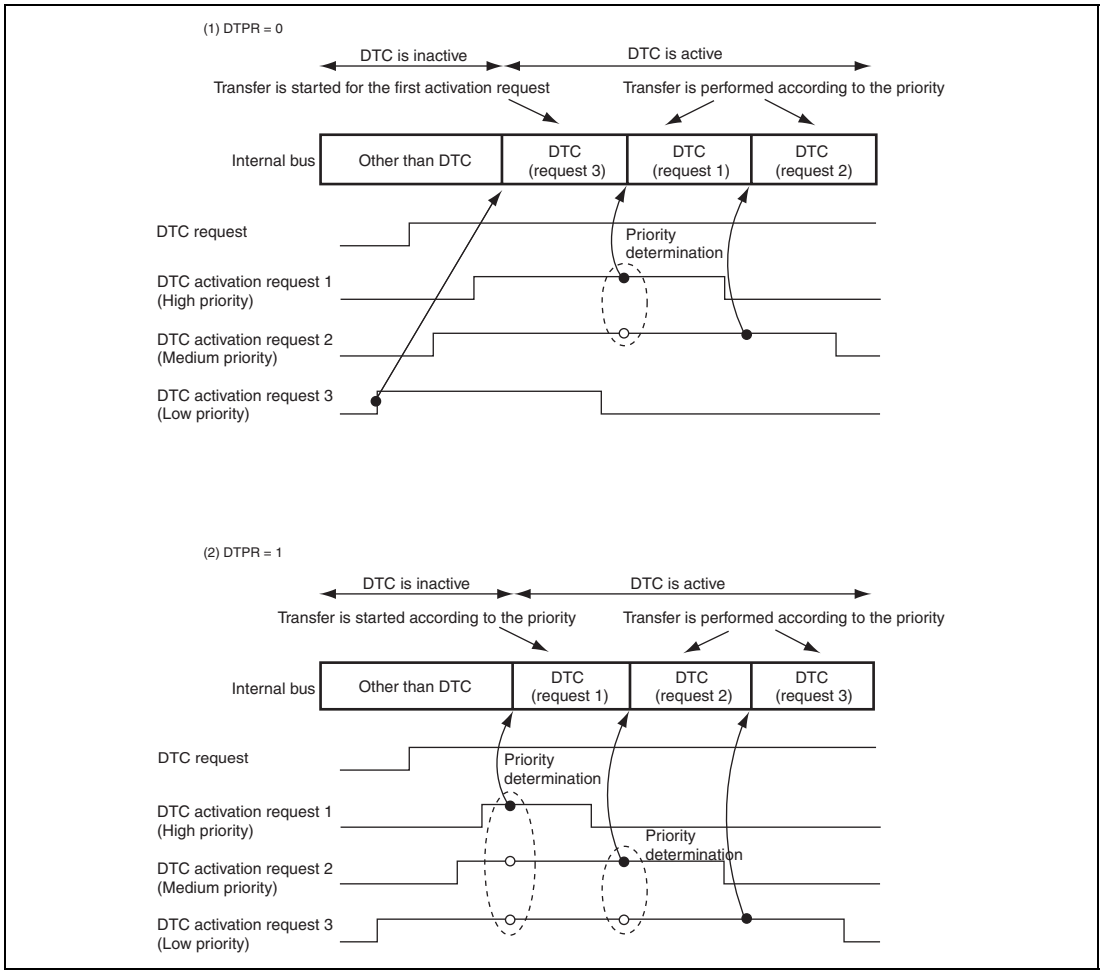
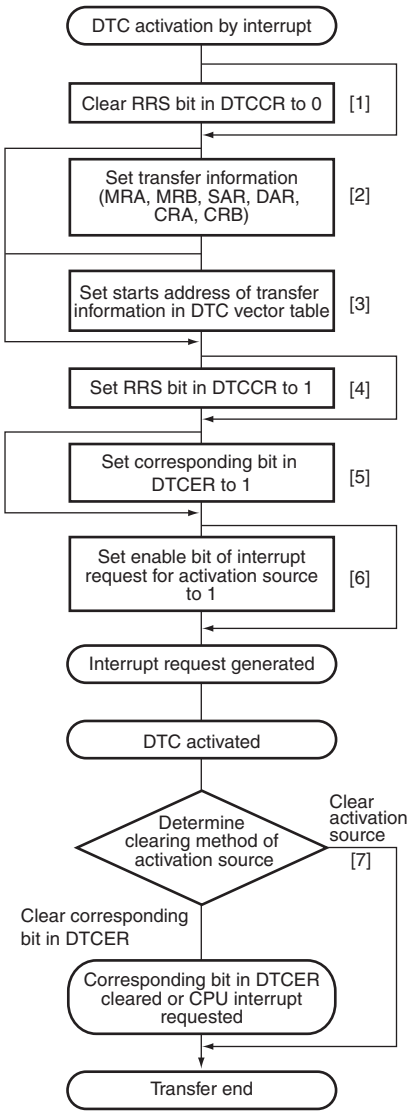


Figure 8.17 Example of DTC Activation According to Priority

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 SCSCR 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 SCSSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from SCRDR 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.

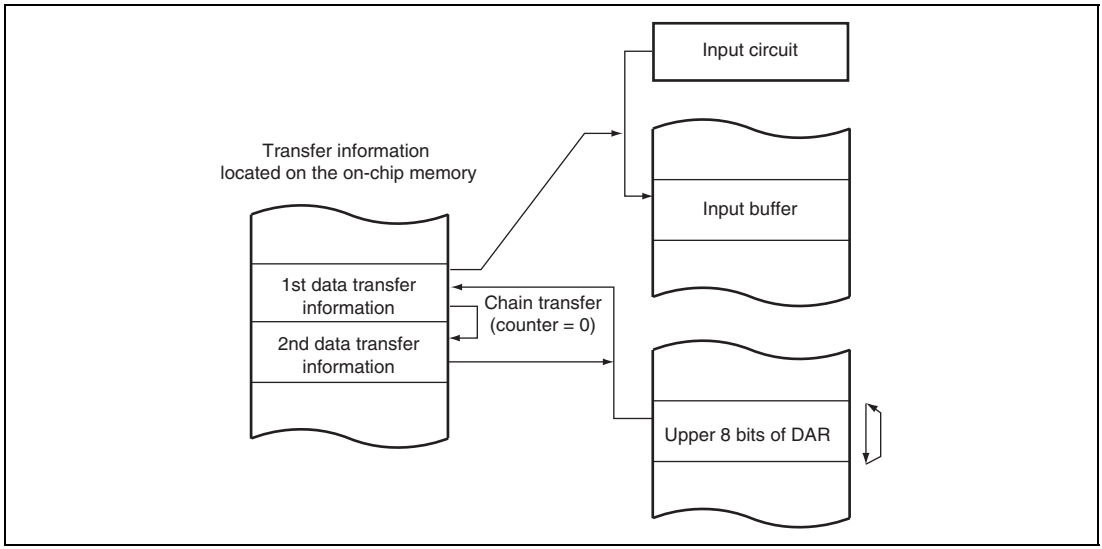


Figure 8.19 Chain Transfer when Transfer Counter = 0

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, Interrupt Requests.

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 26, 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 when write 1 to it. Furthermore, the DTCE bit can be changed after confirming that flags related to stopping of the activating source for the DTC are not set.

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, 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

When the request from an interrupt source acting as an activating source for the DTC is conveyed to the CPU as an interrupt on completion of DTC transfer, the interrupt source flag should be cleared from within the interrupt handler as is usually the case. 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 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.

8.9.12 Note on Writing to DTCER

When making settings for a request from an interrupt source acting as an activating source for the DTC to be conveyed to the CPU as an interrupt on completion of DTC transfer, if the given interrupt is generated while the settings are being made by the DTCER register, the DTC and CPU may be activated at the same time. Determine the value of DTCER register before allowing the generation of DTC activation interrupts.

Section 9 Bus State Controller (BSC) (SH7239A and SH7237A only)

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 and other memory storage devices, and external devices.

9.1 Features

1. External address space
   - A maximum of 2 Mbytes for each of areas CS0, CS1, CS3 to CS6.
   - Can specify the normal space interface and MPX-I/O for each address space.
   - Can select the data bus width (8 or 16 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. MPX-I/O interface
   - Can directly connect to a peripheral LSI that needs an address/data multiplexing.
4. Bus arbitration
   - Shares all of the resources with other CPU and outputs the bus enable after receiving the bus request from external devices.

Figure 9.1 shows a block diagram of the BSC.

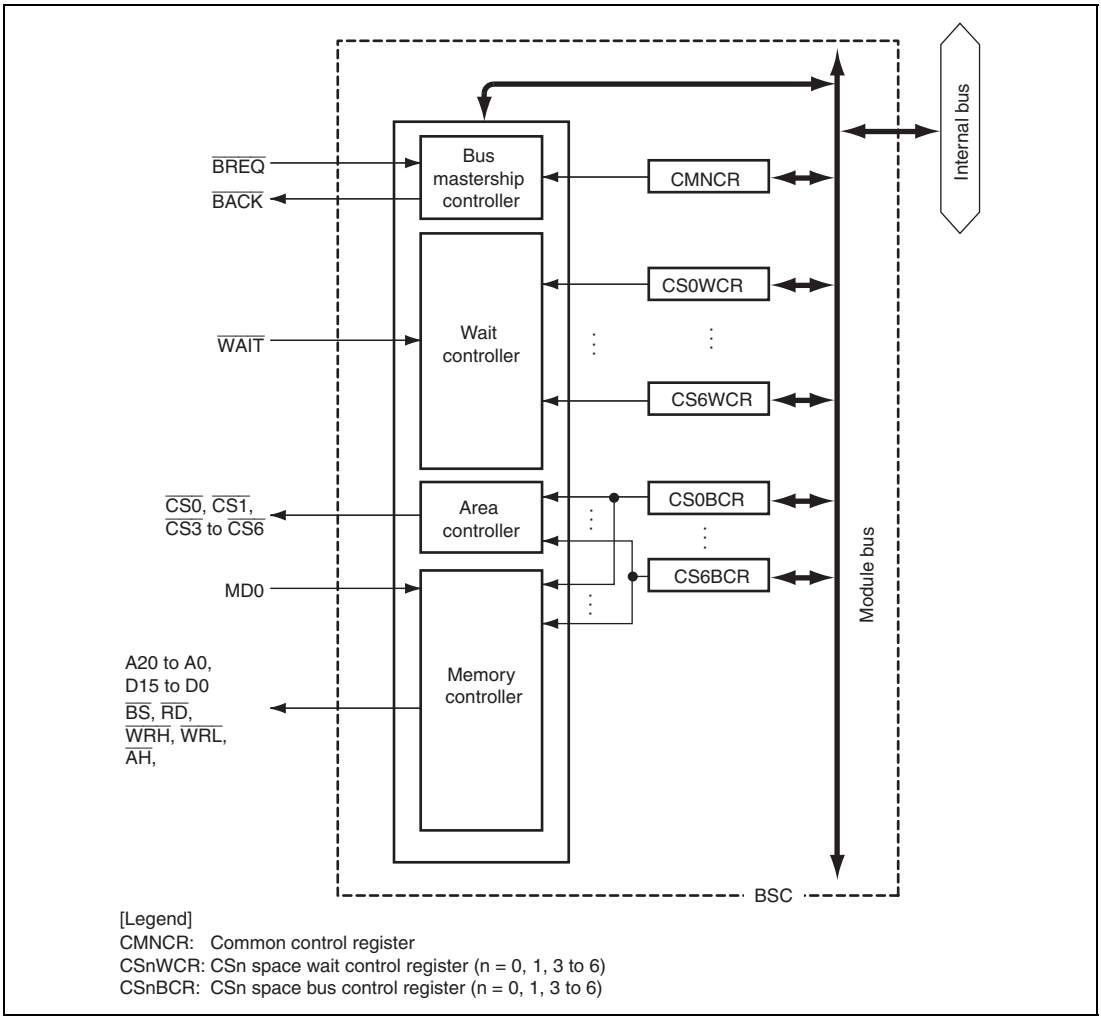


Figure 9.1 Block Diagram of BSC

9.2 Input/Output Pins

Table 9.1 shows the pin configuration of the BSC.

Table 9.1 Pin Configuration

Name	I/O	Function
A20 to A0	Output	Address bus
D15 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ to $\overline{CS6}$	Output	Chip select
$\overline{RD}$	Output	Read pulse signal (read data output enable signal)
$\overline{AH}$	Output	A signal used to hold an address when MPX-I/O is in use
$\overline{WRH}$	Output	Indicates that D15 to D8 are being written to.
$\overline{WRL}$	Output	Indicates that D7 to D0 are being written to.
$\overline{WAIT}$	Input	External wait input
$\overline{BREQ}$	Input	Bus request input
$\overline{BACK}$	Output	Bus enable output

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 space. For details of the external address space, see section 3, MCU Operating Modes.

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. For details of mode settings, see section 3, MCU Operating Modes.

- 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

  In on-chip ROM-enabled mode, since 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. Area 0 is the external memory space.

- Initial Settings of Data Bus Widths for Areas 0, 1, 3 to 6

  The initial settings of data bus widths of areas 0, 1, 3 to 6 can be selected at a time as 8 bits or 16 bits.

  In on-chip ROM-enabled mode, all the data bus widths of areas 0, 1, 3 to 6 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 0, 1, 3 to 6 can be selected as big endian or little endian. In on-chip ROM-enabled mode, all the endianness of areas 0, 1, 3 to 6 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.

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 9.2 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'36DB0200*	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
Bus function extending register	BSCEHR	R/W	H'0000	H'FFFE3C1A	16

Note: * This value is when selecting the 8-bit bus width, whereas the value is H'36DB0400 when selecting the 16-bit bus width.

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	-
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

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 A20 to A0, $\overline{BS}$, $\overline{CSn}$, $\overline{WRH}$, $\overline{WRL}$, $\overline{AH}$, 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	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0, 1, 3 to 6)

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'36DB0200 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 9.5.6, 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: Reserved (setting prohibited) 010: MPX-I/O 011: Reserved (setting prohibited) 100: Reserved (setting prohibited) 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited) 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: Reserved (setting prohibited) For MPX-I/O, selects bus width by address. Note: 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.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

9.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0, 1, 3 to 6)

CSnWCR specifies various wait cycles for memory access. 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, MPX-I/O

- CS0WCR

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
	-	-	-	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 13	—	All 0	R/W	Reserved These bits should be 0 when the normal space is selected.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS0}$ Assertion to $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Assertion Specify the number of delay cycles from address and $\overline{CS0}$ assertion to $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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 $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ negation to address and $\overline{CS0}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS1WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	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	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 19	—	All 0	R	Reserved These bits are 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, $\overline{CS1}$ Assertion to $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Assertion Specify the number of delay cycles from address and $\overline{CS1}$ assertion to $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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)
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

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	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Negation to Address, $\overline{CS1}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ negation to address and $\overline{CS1}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- 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
	-	-	-	-	-	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 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
	-	-	-	-	-	-	-	-	-	-	-	-	-	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	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 19	—	All 0	R	Reserved These bits are 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, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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 $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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	-	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.
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: Setting prohibited The relationship between the SZSEL bit and bus width selected by A14 is summarized below: • SZSEL = 0 and A14 = 0: 8-bit bus width • SZSEL = 0 and A14 = 1: 16-bit bus width
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
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 $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Assertion Specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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 $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Negation to Address, $\overline{CS5}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ negation to address and $\overline{CS5}$ negation. 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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	-	-	-	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 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 $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Assertion Specify the number of delay cycles from address, $\overline{CS6}$ assertion to $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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 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 $\overline{RD}$, $\overline{WRH}$, $\overline{WRL}$ Negation to Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ negation to address, and $\overline{CS6}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

9.4.4 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, 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 IRQ6 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. Setting DTLOCK to 1 is prohibited.

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 0, 1, 3 to 6, endian can be set by the CSnBCR setting while the target space is not accessed.

For normal memory, the data bus width can be selected from two widths (8 and 16 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 9.3 to 9.6 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 9.3 16-Bit External Device Access and Data Alignment in Big-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{WRH}$**	**$\overline{WRL}$**
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 9.4 8-Bit External Device Access and Data Alignment in Big-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{WRH}$**	**$\overline{WRL}$**
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 9.5 16-Bit External Device Access and Data Alignment in Little-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{WRH}$**	**$\overline{WRL}$**
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 9.6 8-Bit External Device Access and Data Alignment in Little-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{WRH}$**	**$\overline{WRL}$**
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. 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.

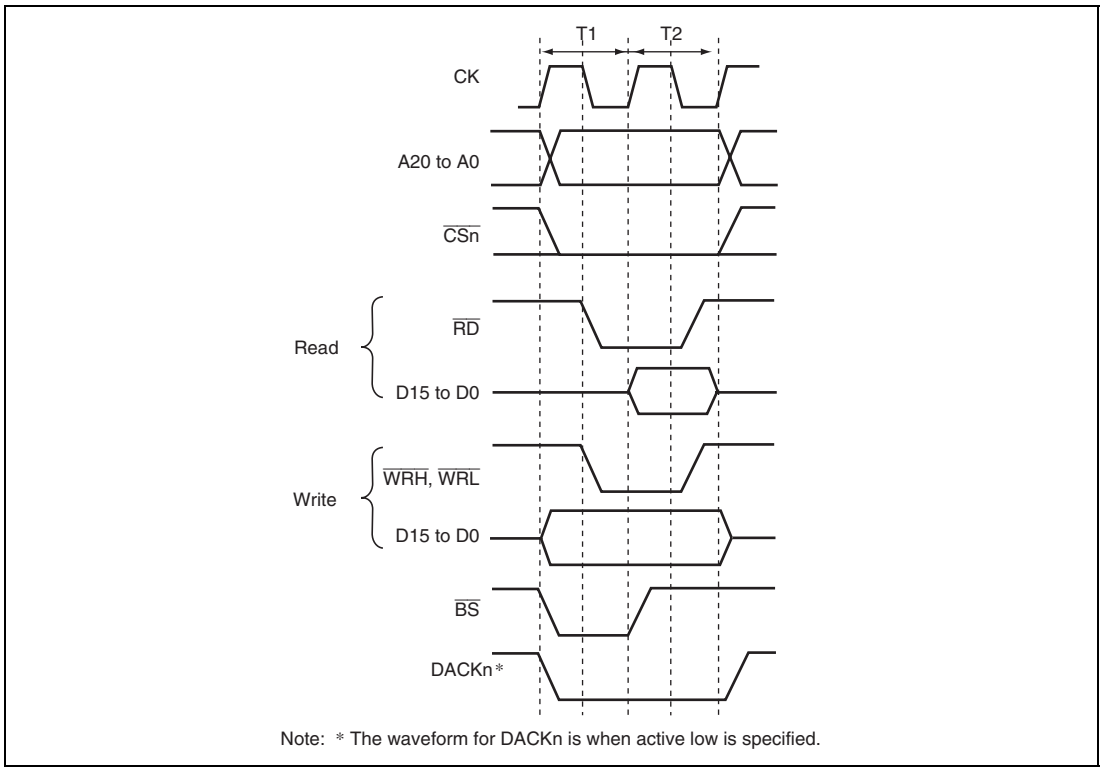


Figure 9.2 Normal Space Basic Access Timing (Access Wait 0)

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 $\overline{WRH}$/$\overline{WRL}$ signal for the byte to be written is asserted.

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).

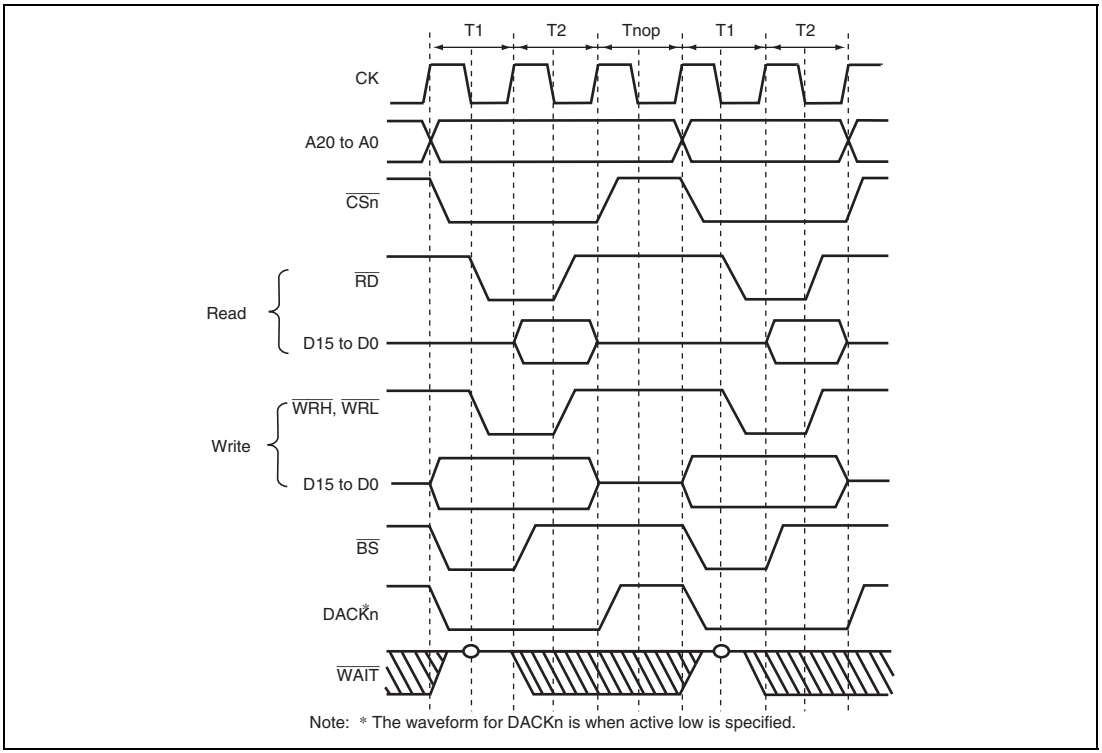


Figure 9.3 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

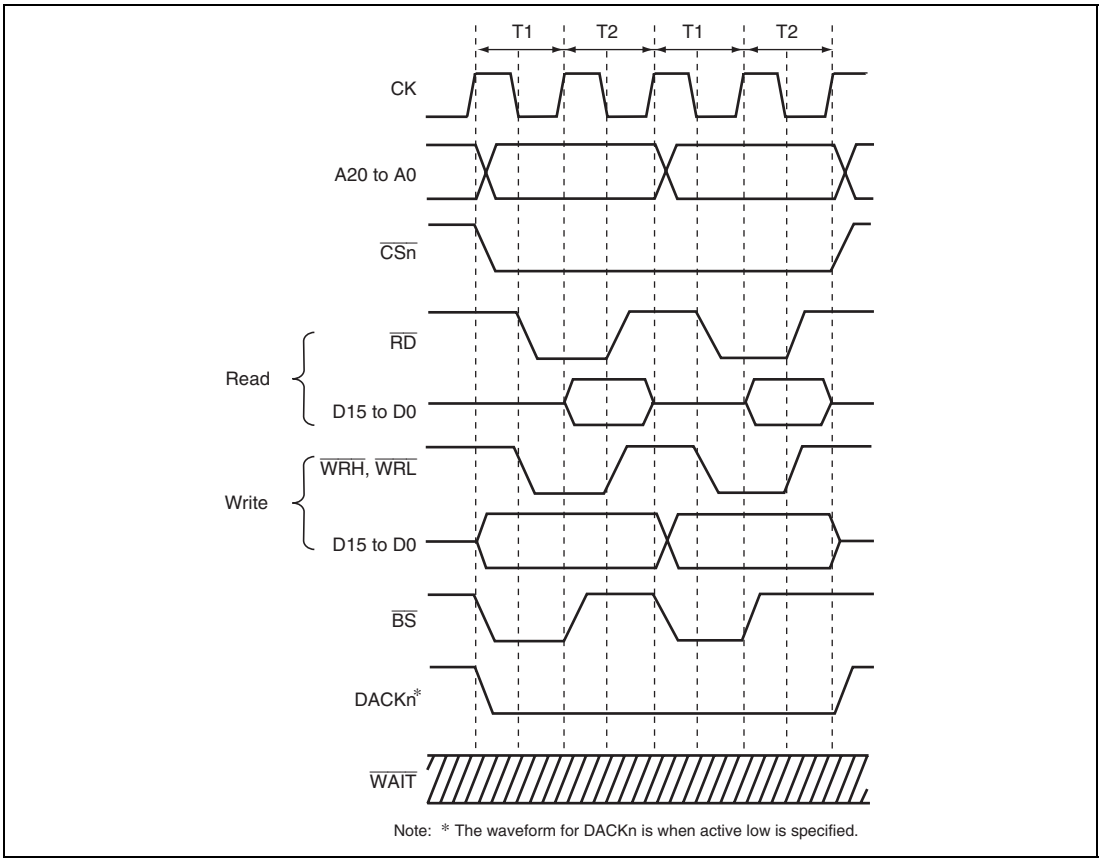


Figure 9.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

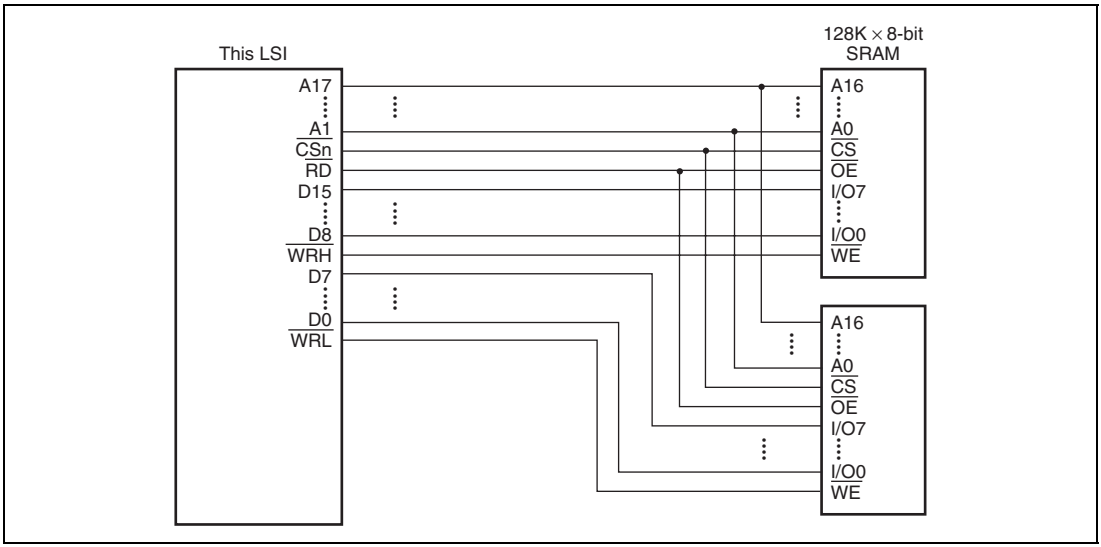


Figure 9.5 Example of 16-Bit Data-Width SRAM Connection

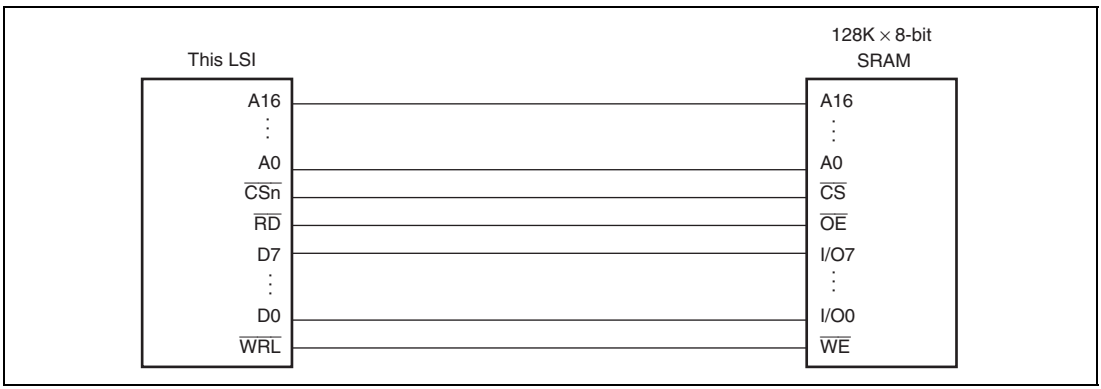


Figure 9.6 Example of 8-Bit Data-Width SRAM Connection

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, and 5 to insert wait cycles independently in read access and in write access. Areas 0, 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 9.7.

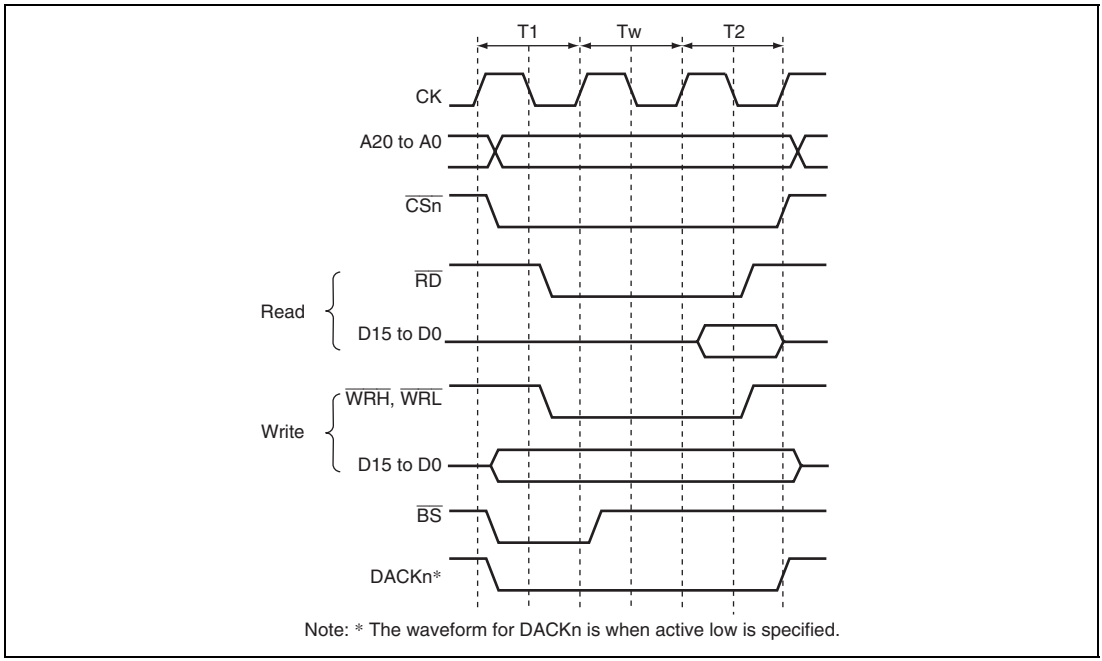


Figure 9.7 Wait Timing for Normal Space Access (Software Wait Only)

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 9.8. 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.

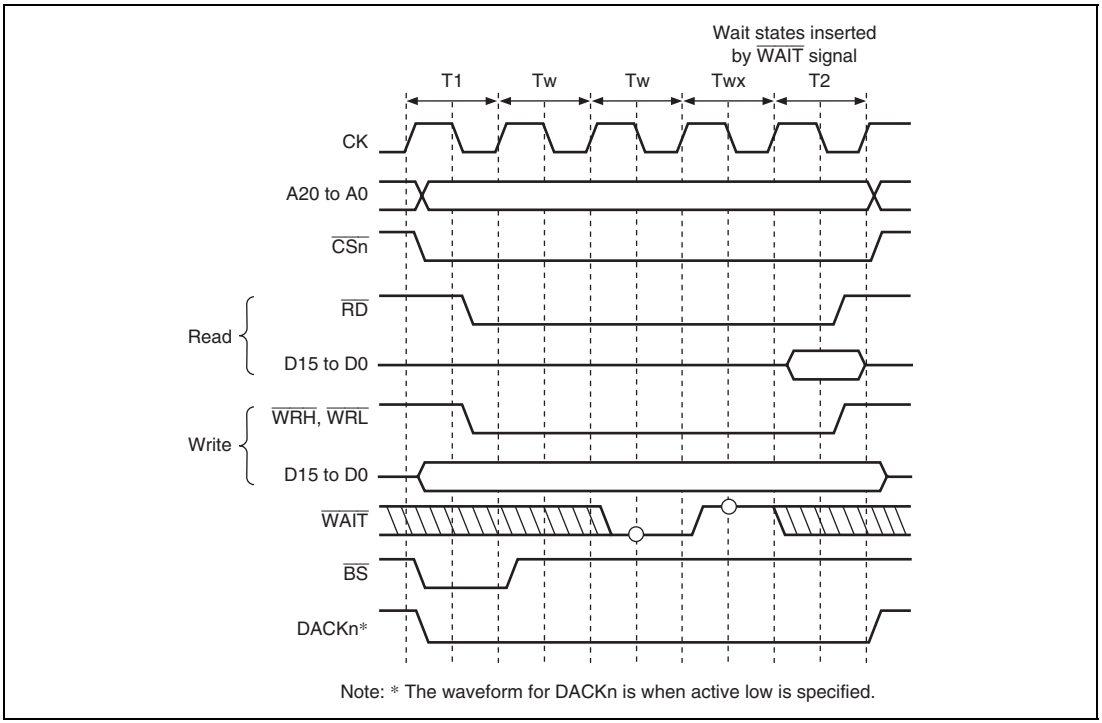


Figure 9.8 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

9.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WRH}}$ and $\overline{\text{WRL}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WRH}}$ and $\overline{\text{WRL}}$ 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 9.9 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 $\overline{\text{WRH}}$ and $\overline{\text{WRL}}$ 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.

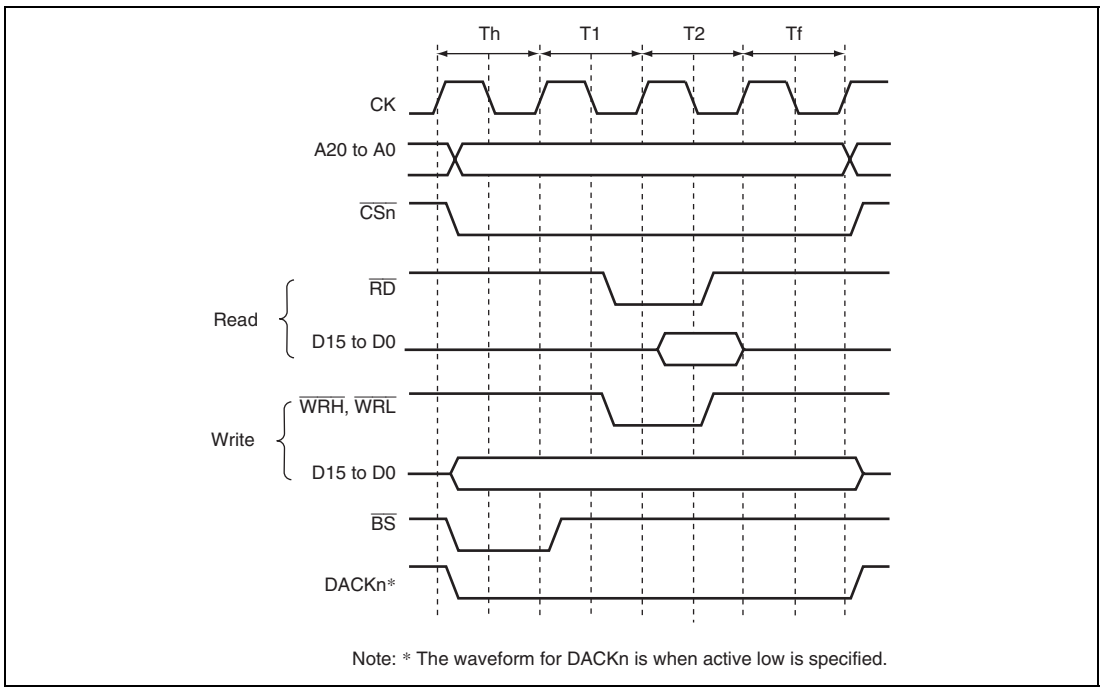


Figure 9.9 $\overline{\text{CSn}}$ Assert Period Expansion

9.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, $\overline{WRH}$, and $\overline{WRL}$ 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 data cycle is the same as that in a normal space access.

Timing charts are shown in figures 9.10 to 9.12.

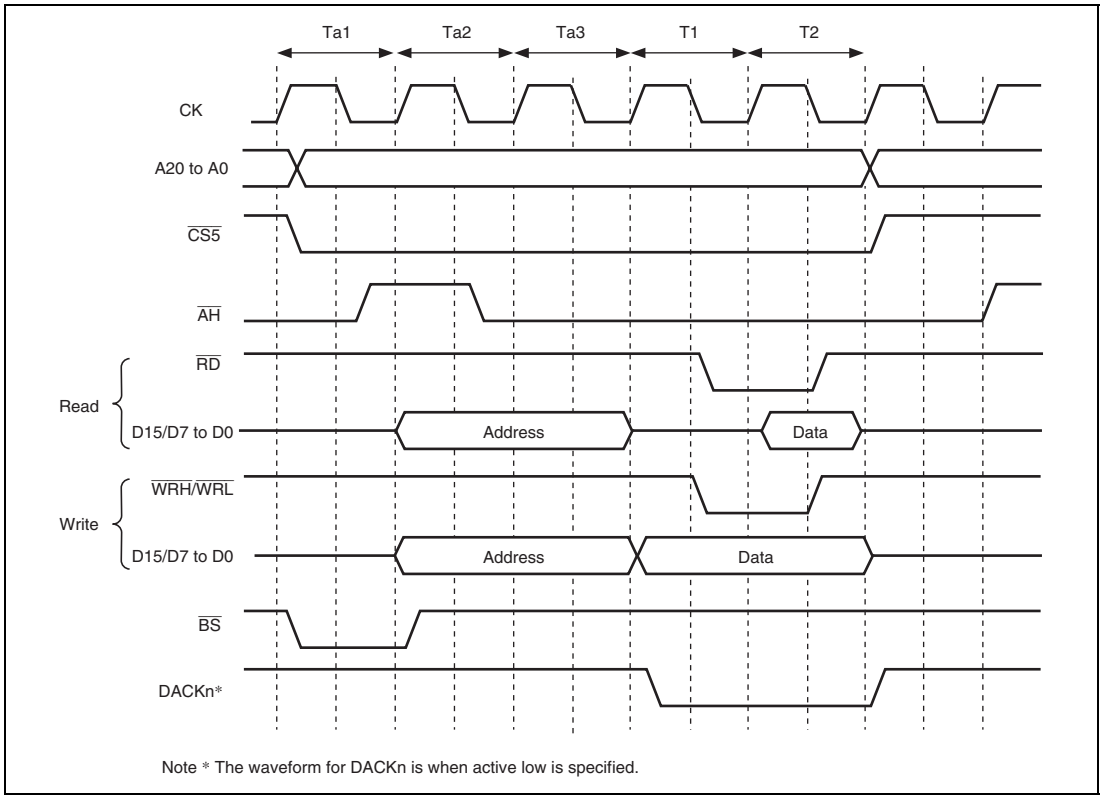


Figure 9.10 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

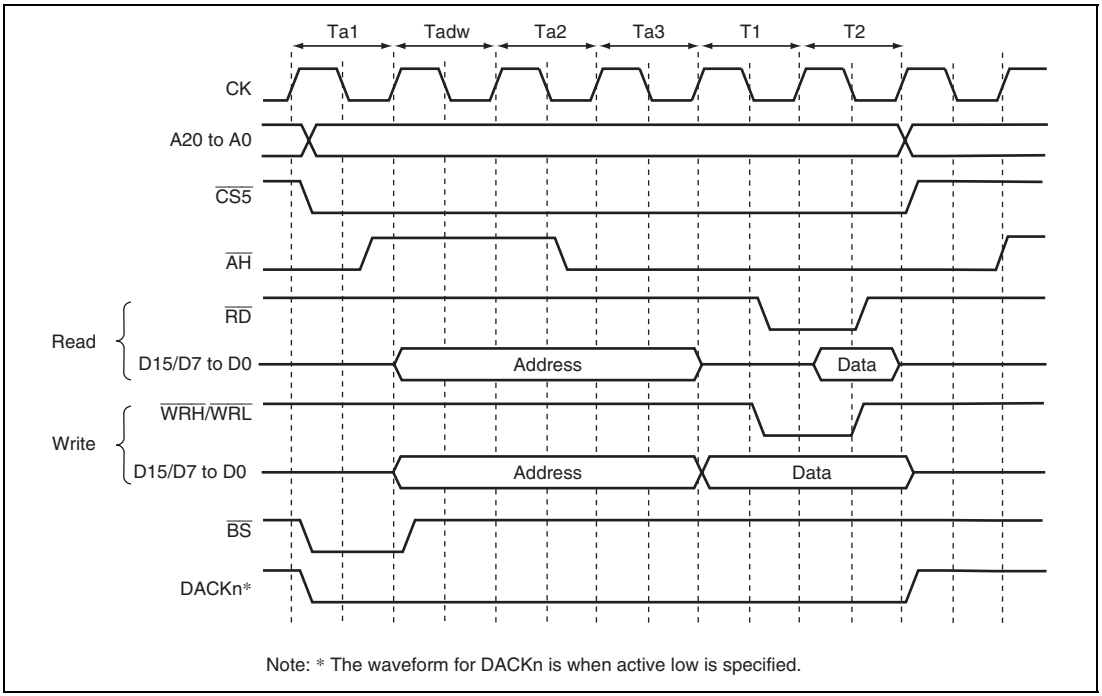


Figure 9.11 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

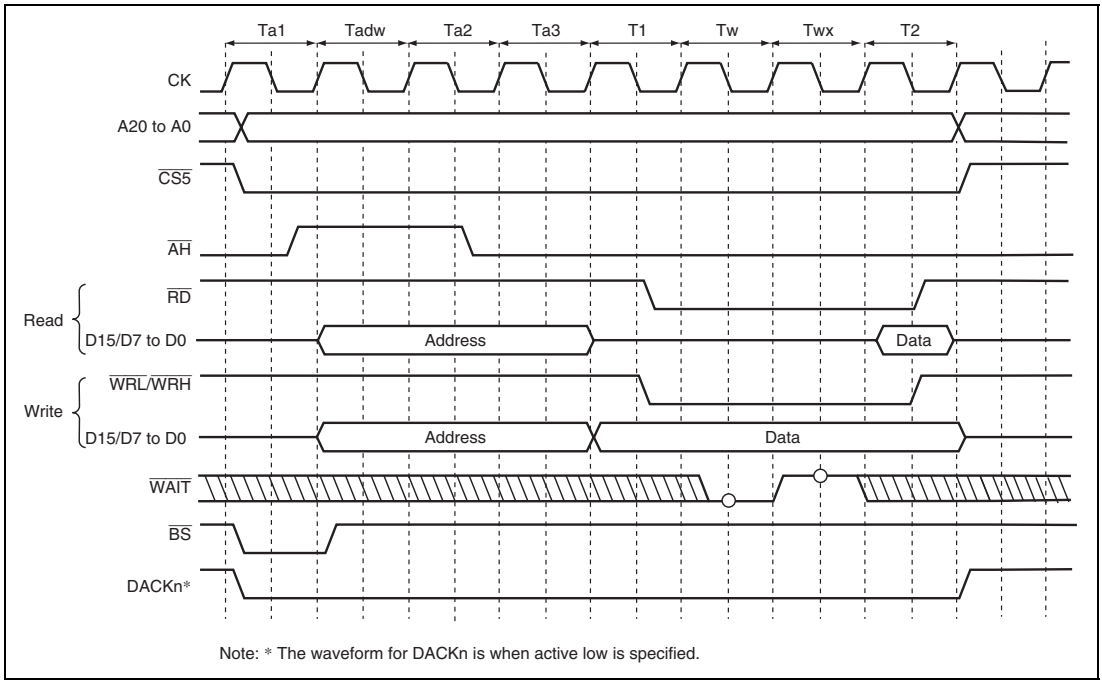


Figure 9.12 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

9.5.6 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 ($\overline{WRH}$, $\overline{WRL}$). 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 $\overline{CSn}$ negation to $\overline{CSn}$ or $\overline{CSm}$ assertion is described below.

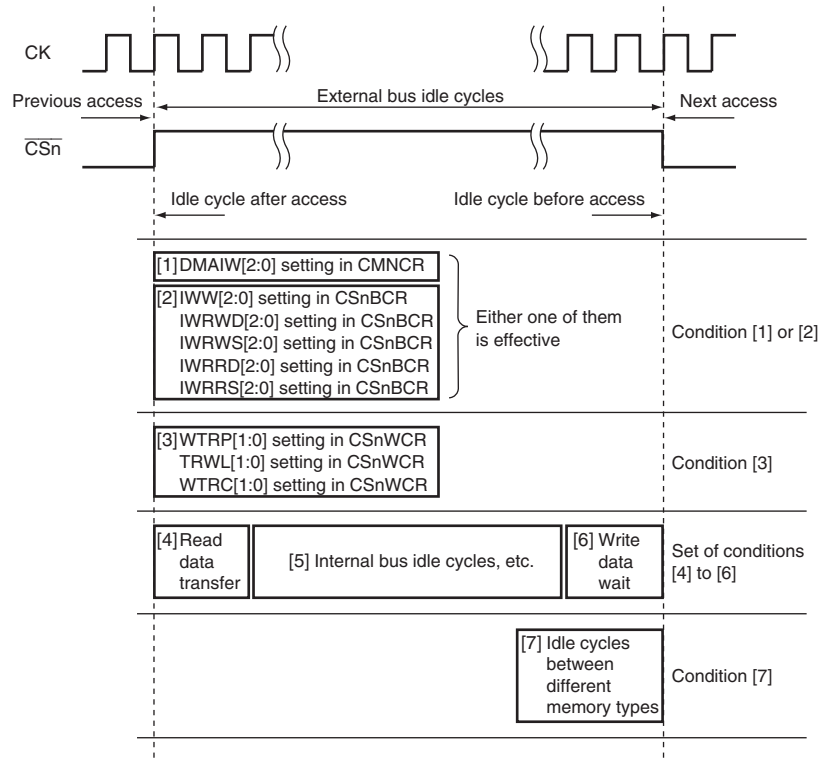
There are seven conditions that determine the number of idle cycles on the external bus as shown in table 9.7. The effects of these conditions are shown in figure 9.13.

Table 9.7 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)	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	
(4)	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	

No.	Condition	Description	Range	Note
(5)	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 9.8 and 9.9 show the relationship between the clock ratio and the minimum number of internal bus idle cycles.
(6)	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.
(7)	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 9.10.

In the above conditions, a total of four conditions, that is, condition (1) or (2) (either one is effective), condition (3), a set of conditions (4) to (6) (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition (7) 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], a set of conditions [4] to [6], and condition [7]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 9.13 Idle Cycle Conditions

Table 9.8 Minimum Number of Idle Cycles on Internal Bus (CPU Operation)

	Clock Ratio (Iϕ:Bϕ)		
CPU Operation	**4:1**	**2:1**	**1:1**
Write → write	2	2	3
Write → read	0	0	1
Read → write	2	2	3
Read → read	0	0	1

Table 9.9 Minimum Number of Idle Cycles on Internal Bus (DMAC Operation)

	Transfer Mode	
DMAC Operation	**Dual Address**	**Single Address**
Write → write	0	2
Write → read	0 or 2	0
Read → write	0	0
Read → read	0	2

Notes: 1. The write → write and read → read columns in dual address transfer indicate the cycles in the divided access cycles.
2. For the write → read cycles in dual address transfer, 0 means different channels are activated successively and 2 means when the same channel is activated successively.
3. 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 9.10 Number of Idle Cycles Inserted between Access Cycles to Different Memory Types

	Next Cycle	
Previous Cycle	**SRAM**	**MPX-I/O**
SRAM	0	1
MPX-I/O	1	0

Figure 9.14 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 CS3 space by CPU access. Transfer is repeated in the following order: CS1 read → CS1 read → CS3 write → CS3 write → CS3 read → ...

- Conditions
  The bits for setting the idle cycles between access cycles in CS1BCR and CS3BCR are all set to 0.
  In CS1WCR and CS3WCR, 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 CS3 spaces, normal SRAM devices are connected, the bus width is 16 bits, and access size is also 16 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]	0	0	0	0	The WM bit is set to 1.
[4]	1	1	0	0	Generated after a read cycle.
[5]	0	2	2	0	See the Iϕ:Bϕ = 4:1 column in table 9.8.
[6]	0	1	0	0	No idle cycle is generated for the second time due to the write buffer effect.
[4] + [5] + [6]	0	4	2	0	
[7]	0	0	0	0	Value for SRAM → SRAM access
Estimated idle cycles	1	4	2	0	Maximum value among conditions [1] or [2], [3], [4] + [5] + [6], and [7]
Actual idle cycles	1	4	2	1	The estimated value does not match the actual value in the W → R cycles because the internal idle cycles due to condition [5] is estimated as 0 but actually an internal idle cycle is generated due to execution of a loop condition check instruction.

Figure 9.14 Comparison between Estimated Idle Cycles and Actual Value

9.5.7 Bus Arbitration

The bus arbitration of this LSI has the bus mastership in the normal state and releases the bus mastership after receiving a bus request from another device.

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 $\overline{\text{CSn}}$ 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.

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 ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{WRH}}$, $\overline{\text{WRL}}$, and $\overline{\text{RD}}$) are placed in the high-impedance state at subsequent rising edges of CK. These bus control signals are driven high at least one cycle before it is placed in the high-impedance state. Bus request signals are sampled at the falling edge of CK.

The sequence for reclaiming the bus mastership from an external device is described below. 1.5 cycles after the negation of $\overline{\text{BREQ}}$ 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 9.15 shows the bus arbitration timing.

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 $\overline{\text{BREQ}}$ input signal is ignored in standby mode and the $\overline{\text{BACK}}$ 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 $\overline{\text{BACK}}$ pin to enter standby mode.

The bus mastership release ($\overline{\text{BREQ}}$ signal for high level negation) after the bus mastership request ($\overline{\text{BREQ}}$ signal for low level assertion) must be performed after the bus usage permission ($\overline{\text{BACK}}$ signal for low level assertion). If the $\overline{\text{BREQ}}$ signal is negated before the $\overline{\text{BACK}}$ signal is asserted, only one cycle of the $\overline{\text{BACK}}$ signal is asserted depending on the timing of the $\overline{\text{BREQ}}$ signal to be negated and this may cause a bus contention between the external device and the LSI.

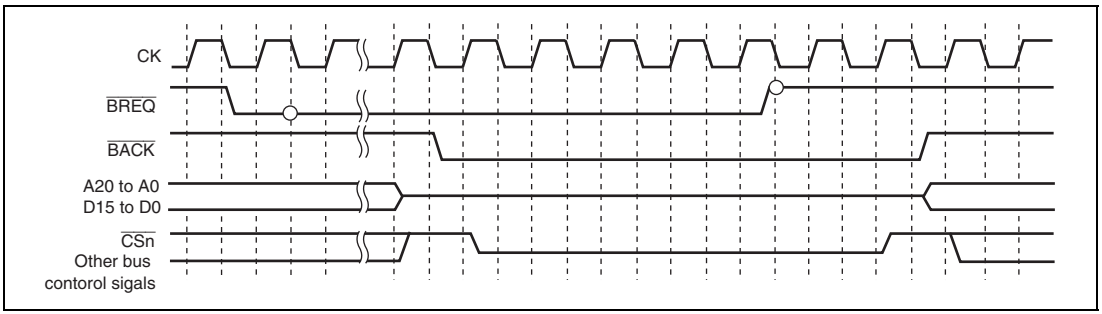


Figure 9.15 Bus Arbitration Timing

9.5.8 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.

(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 9.11 shows the number of cycles required for access to the on-chip peripheral I/O registers by the CPU.

Table 9.11 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
	$(2 + n) \times I\phi + (1 + m) \times B\phi + 3 \times P\phi$	FLD 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
	$(2 + n) \times I\phi + (1 + m) \times B\phi + 3 \times P\phi + (2 + l) \times I\phi$	FLD 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 l = 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 9.16 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 9.16). 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.

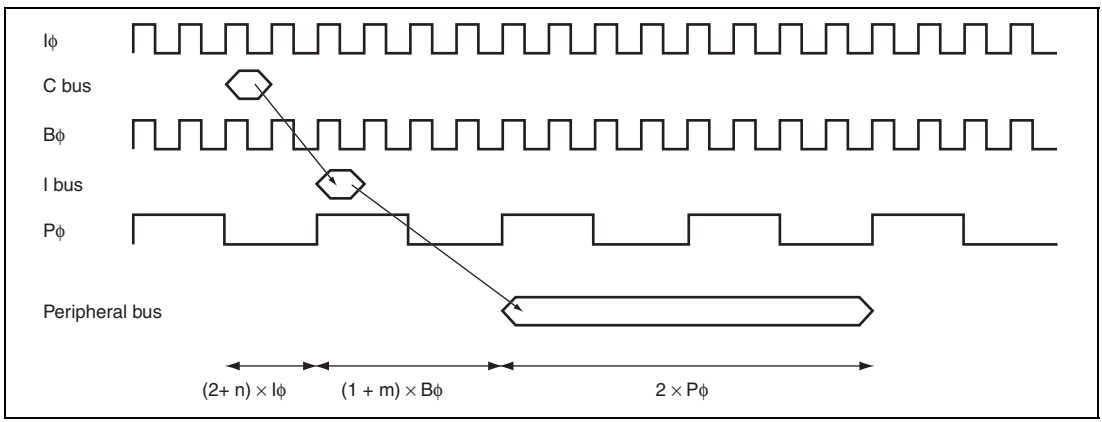


Figure 9.16 Timing of Write Access to On-Chip Peripheral I/O Registers When Iϕ;Bϕ:Pϕ = 4:4:1

Figure 9.17 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 9.17, where n = 1, m = 1, and l = 1, the time required for access is 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 3 × Iϕ.

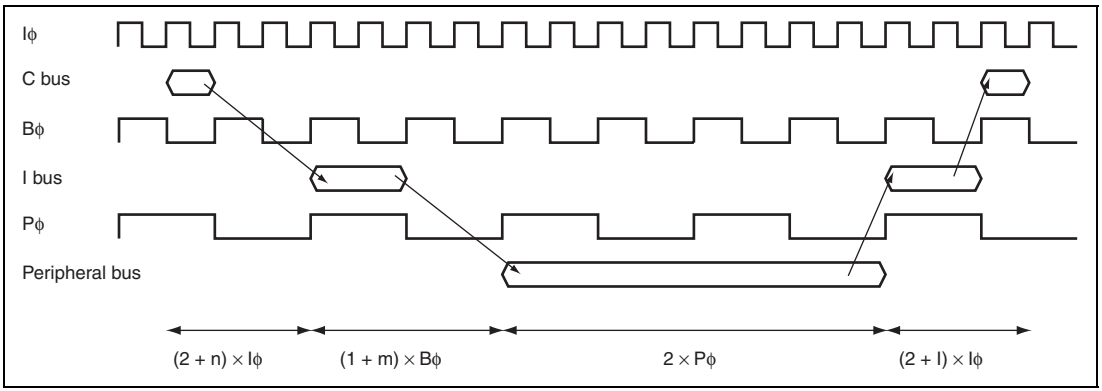


Figure 9.17 Timing of Read Access to On-Chip Peripheral I/O Registers When Iϕ:Bϕ:Pϕ = 4:2:1

9.6 Usage Note

9.6.1 Note on Connection of External LSI Circuits such as SRAMs and ASICs

Among the pins for the SRAM and MPX-I/O control signals, $\overline{BS}$ and $\overline{AH}$ are function multiplexed with A19 and A20, respectively. When an external chip (SRAM, ASIC, etc.) is to be connected to the bus of this LSI, follow the instruction below.

- Use the 19 bits from A0 to A18 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, one A/D converter source, five MTU2 sources, two CMT 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.

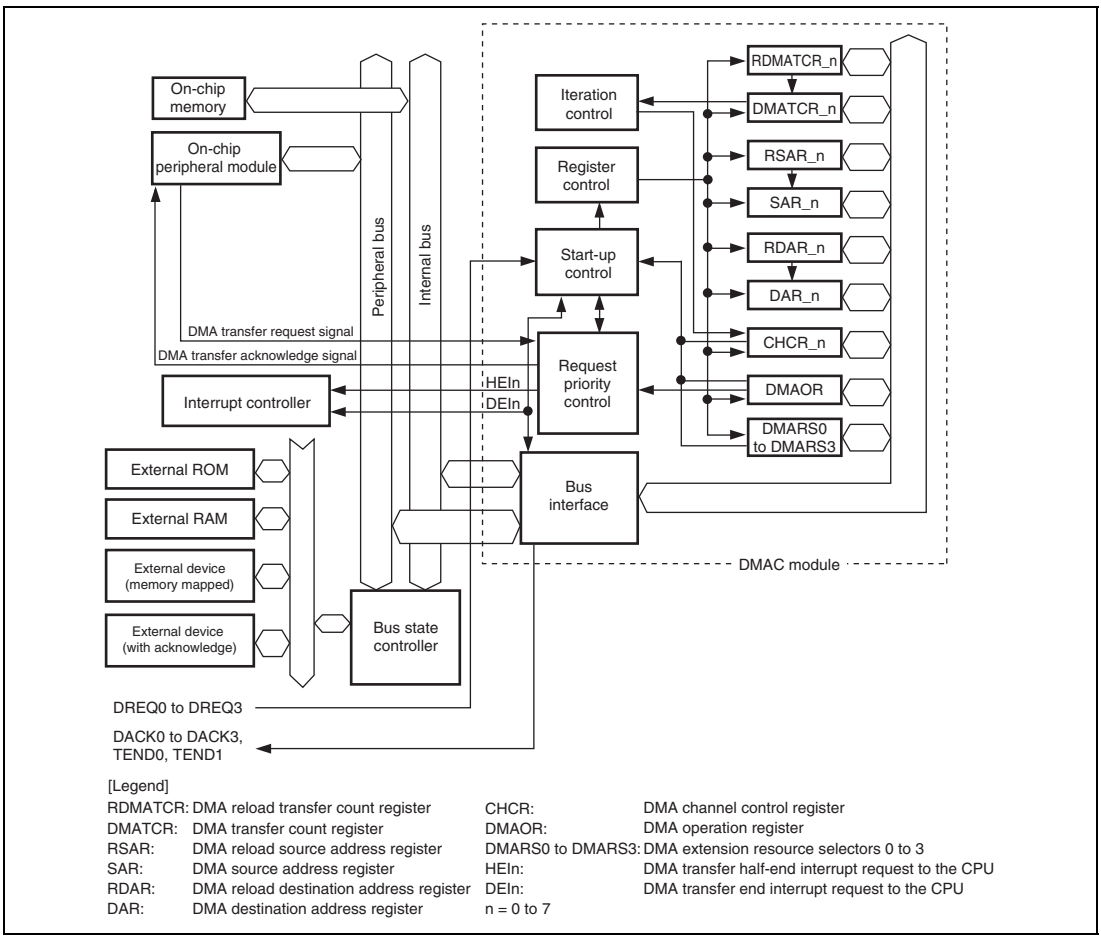


Figure 10.1 Block Diagram of DMAC

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.

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 RSPI or SCIF3 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 (discontinuation was after the same number of transfers as were requested) 1: Detects DREQ by overrun 1 (discontinuation was after one more transfer than the number requested)
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 1: DMA transfer ends by the specified count (DMATCR = 0) [Clearing condition] • Writing 0 after reading TE = 1
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. 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.
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. Write 1 after having read this bit as 0.

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. Only 0 can be written after 1 is read.

If the priority mode bits are modified after a DMA transfer (that is, after the number of transfer operations specified in DMATCR register), the channel priority is initialized. Do not change this while transfer is in progress. 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 SCIF sources, one A/D converter source, five MTU2 sources, two CMT 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
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
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

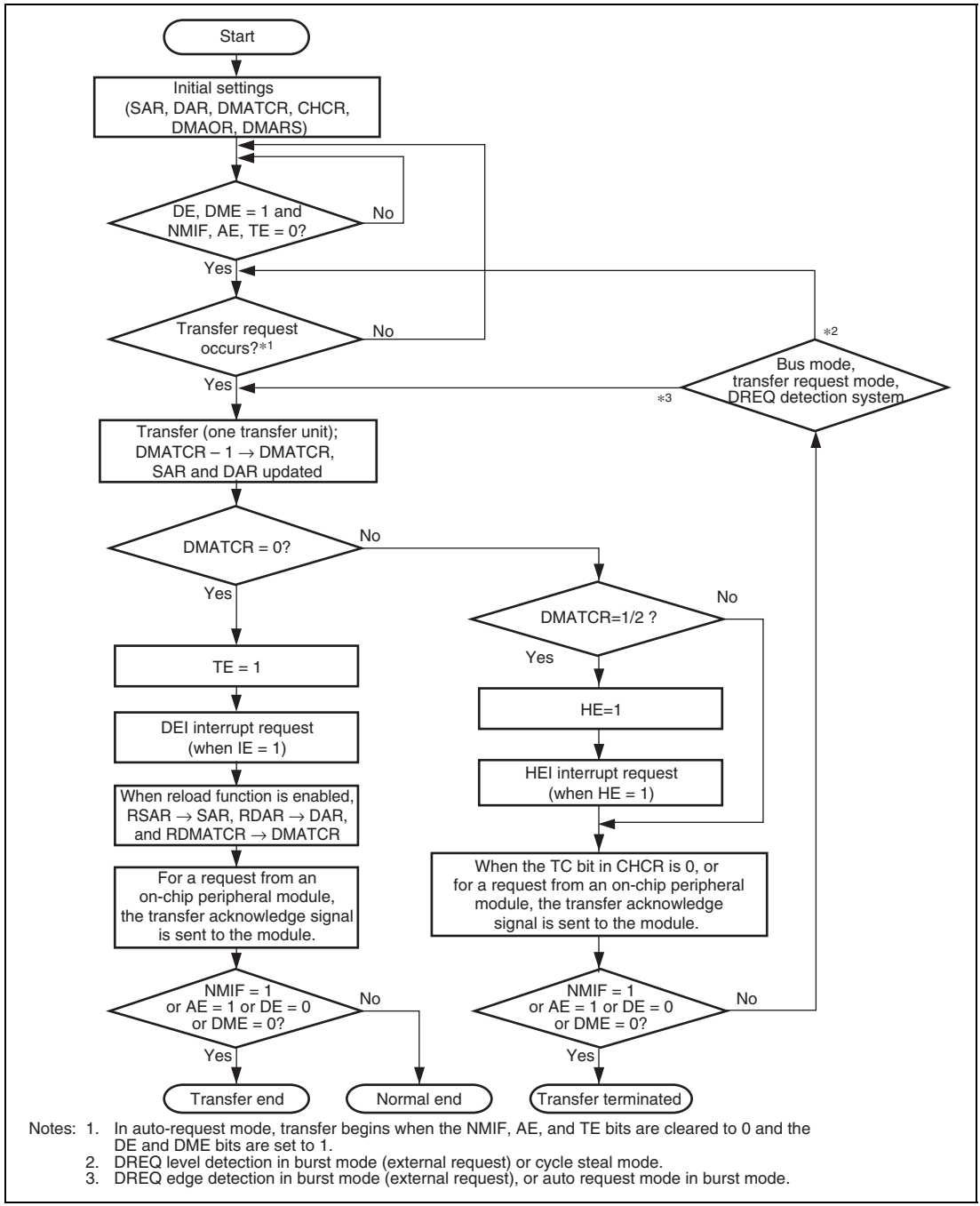
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request. 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 MTU2, 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 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	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	
	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.

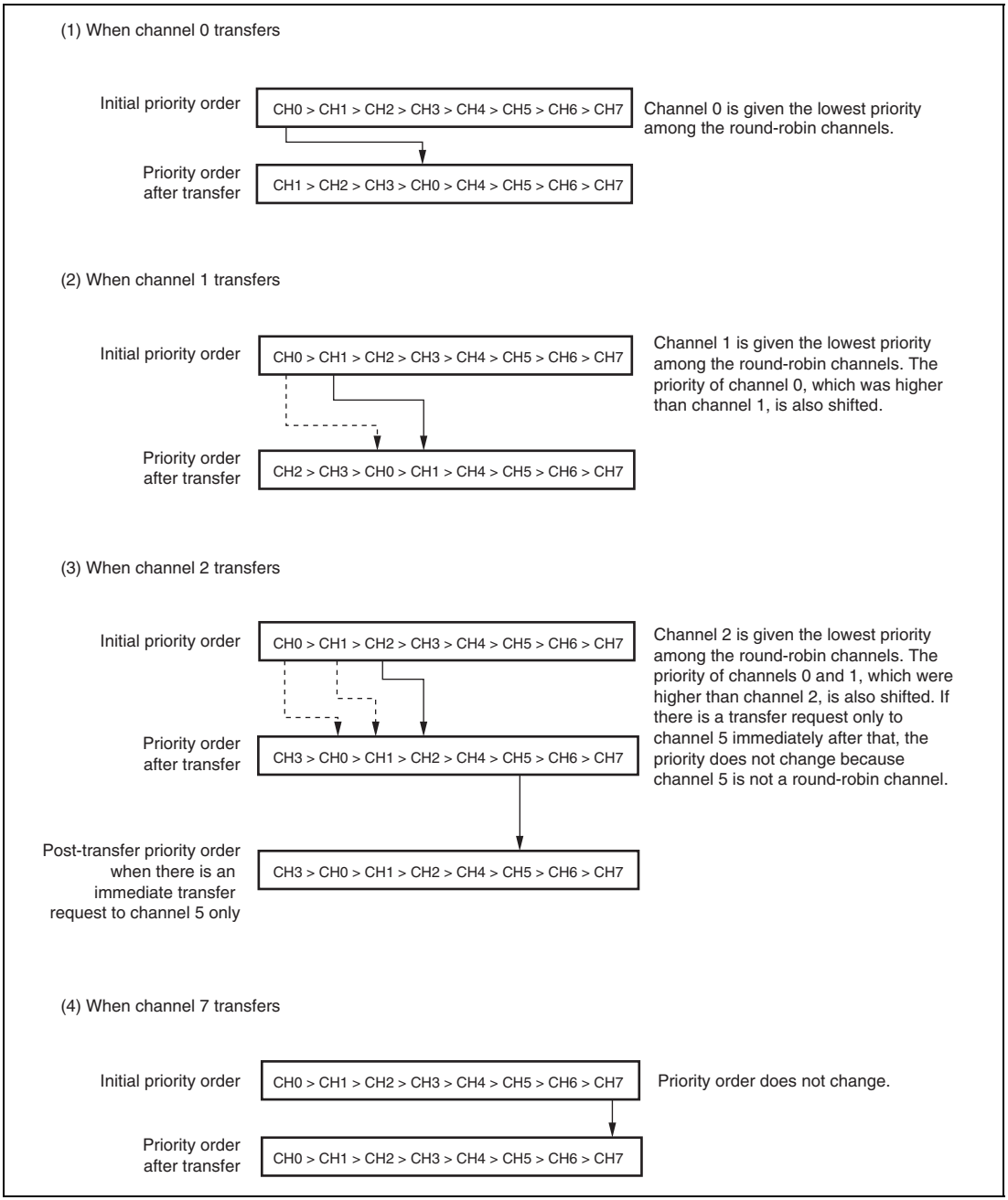


Figure 10.3 Round-Robin Mode

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.

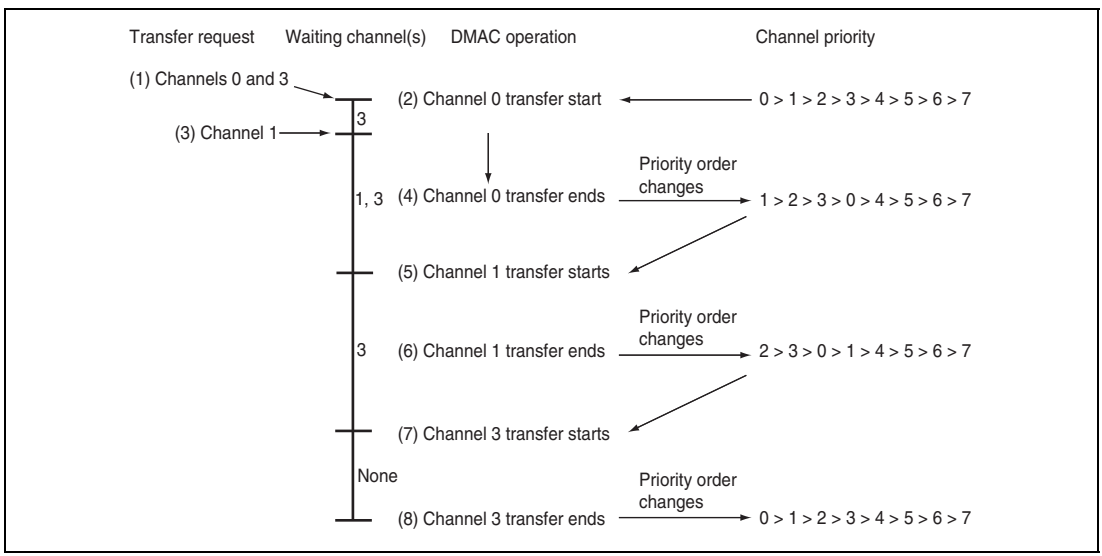


Figure 10.4 Changes in Channel Priority in Round-Robin Mode

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.

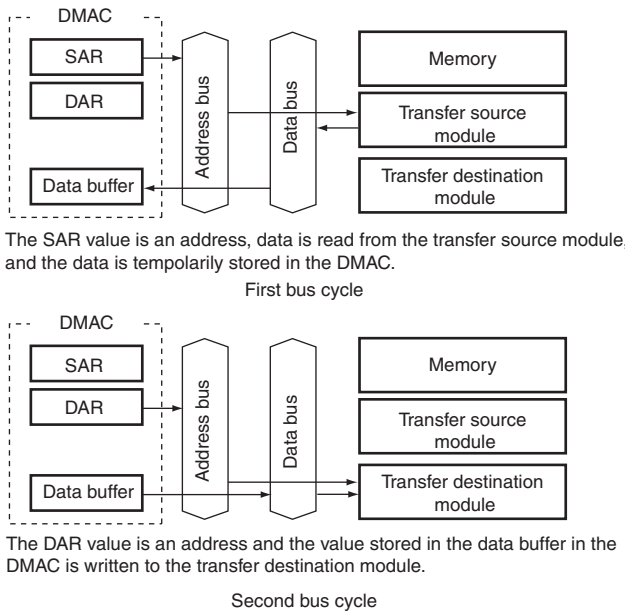
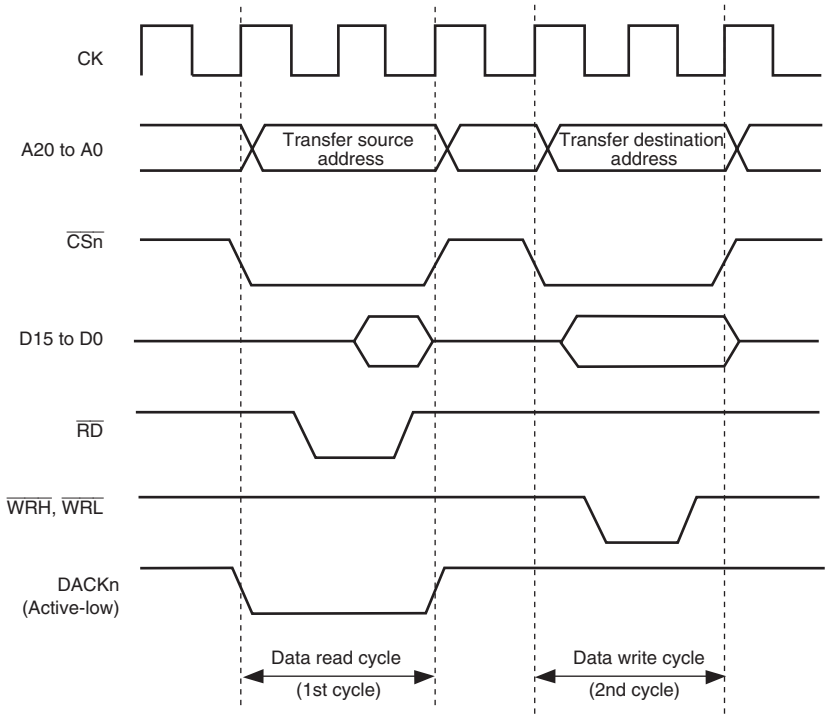


Figure 10.5 Data Flow of Dual Address Mode

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.

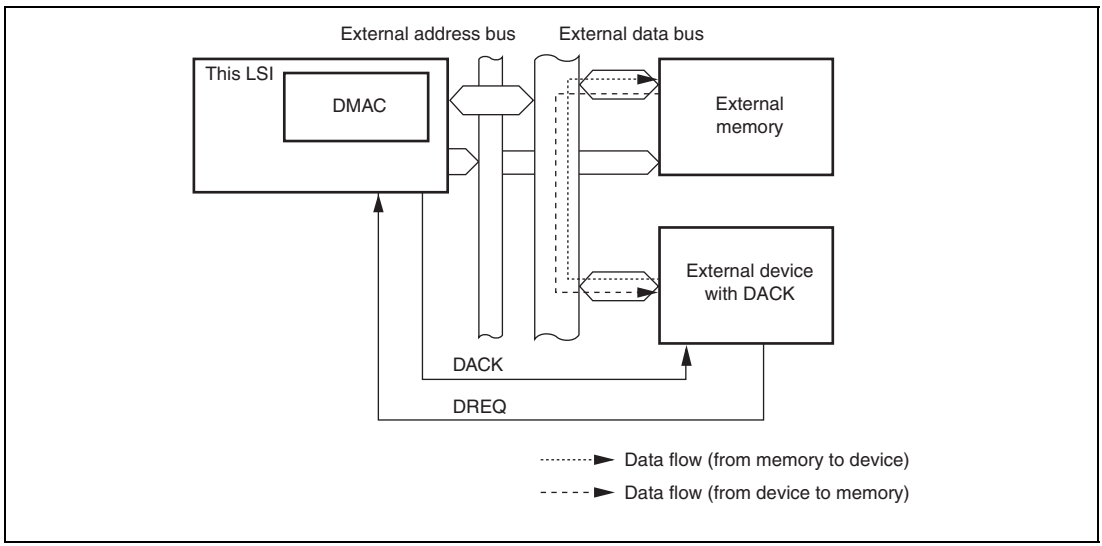
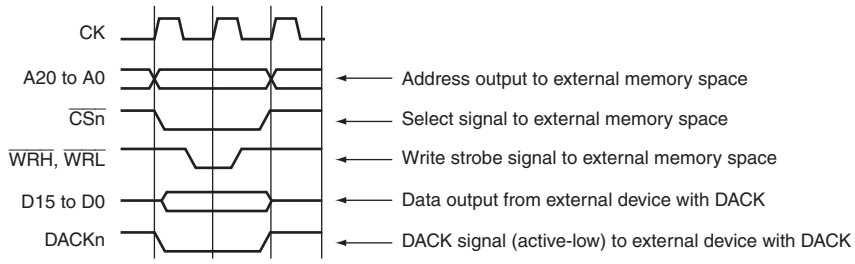


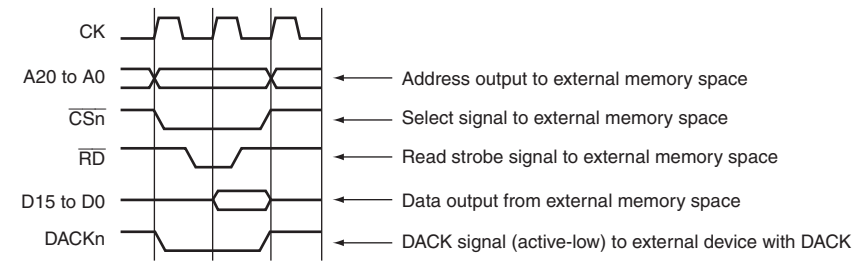
Figure 10.7 Data Flow in Single Address Mode

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

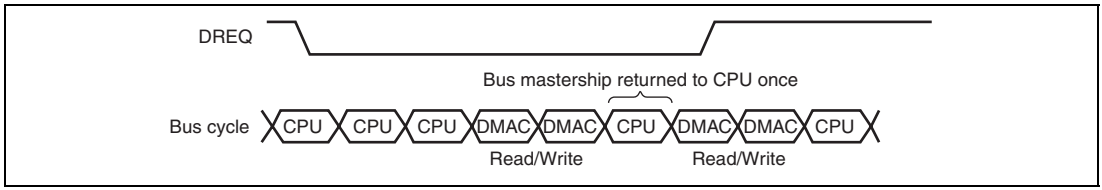


Figure 10.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- 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

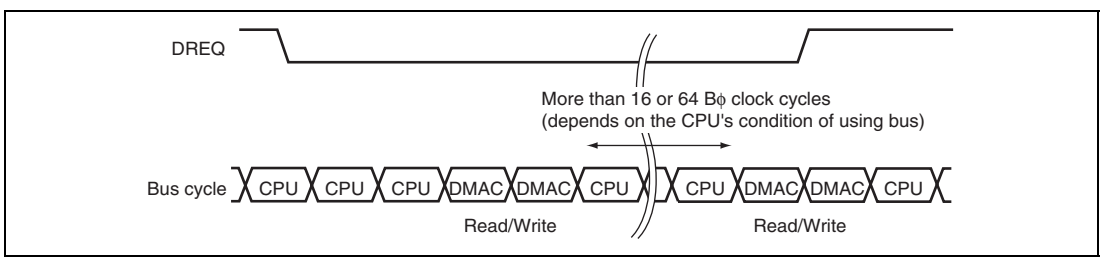


Figure 10.10 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(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.

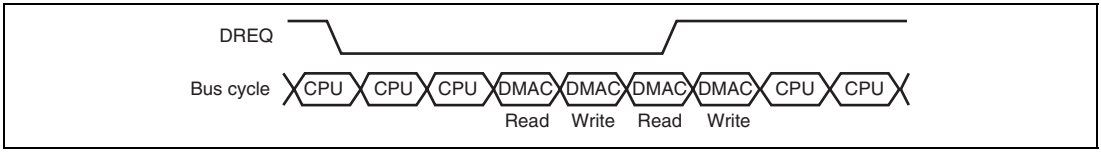


Figure 10.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(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.

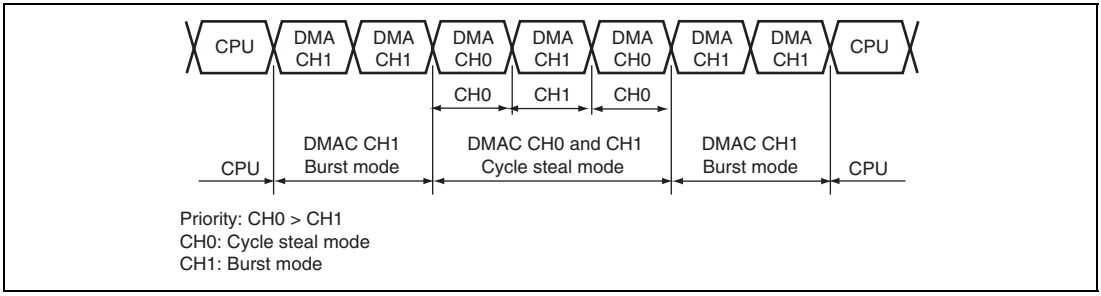


Figure 10.12 Bus State when Multiple Channels are Operating

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) (SH7239A and SH7237A only).

(2) DREQ Pin Sampling Timing

Figures 10.13 to 10.16 show the DREQ input sampling timings in each bus mode.

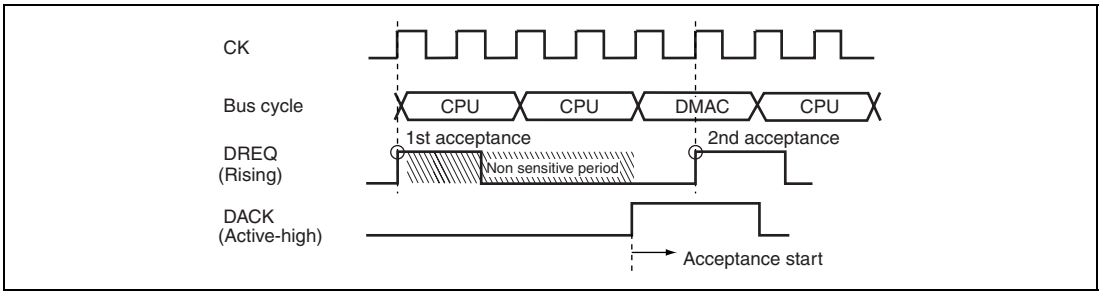


Figure 10.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

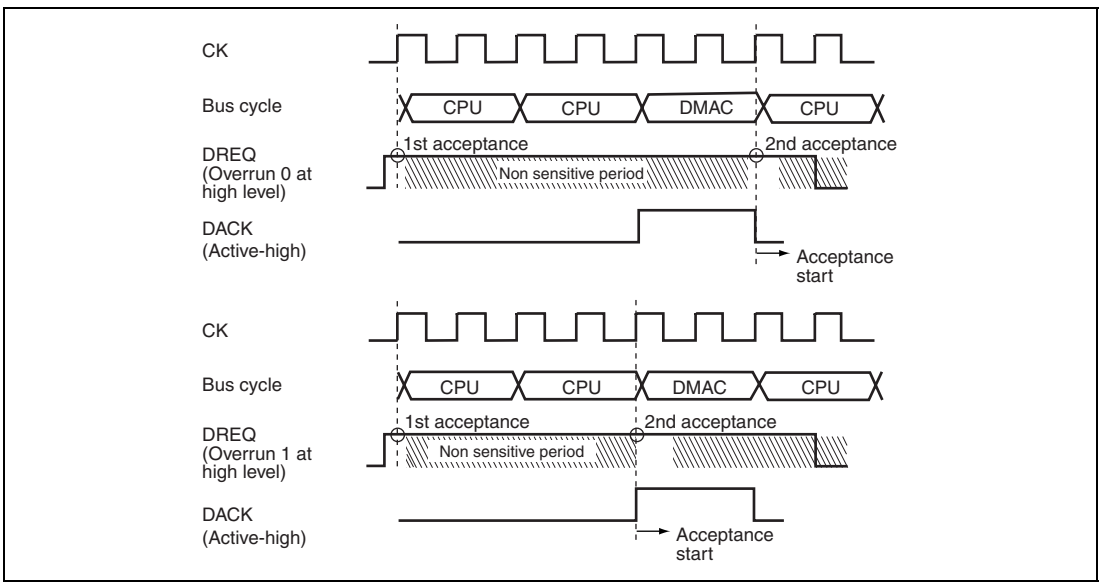


Figure 10.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

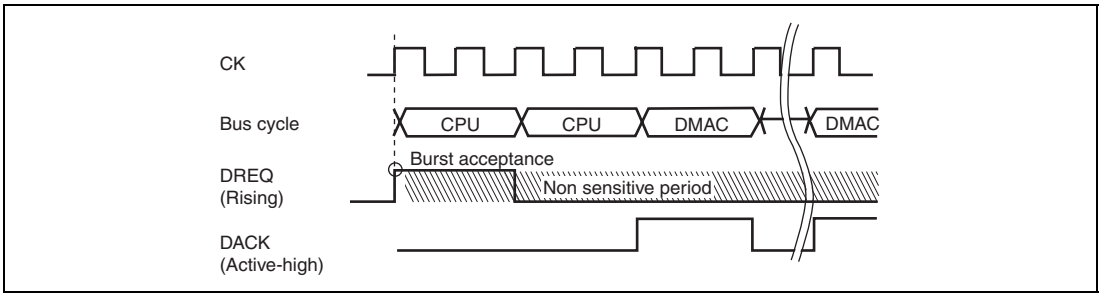


Figure 10.15 Example of DREQ Input Detection in Burst Mode Edge Detection

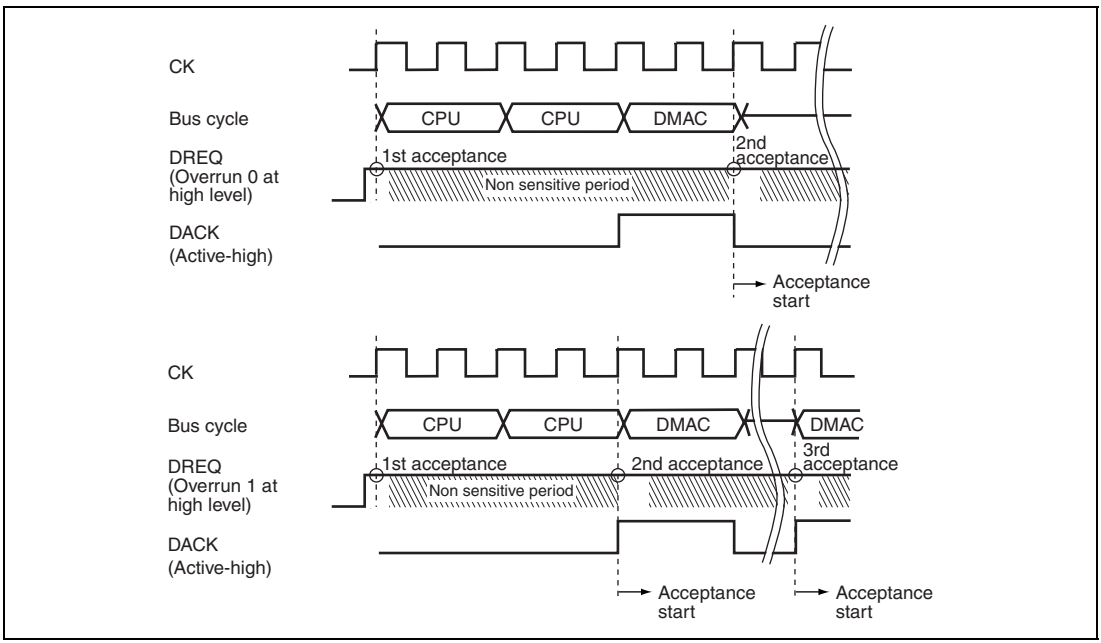


Figure 10.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 10.17 shows the TEND output timing.

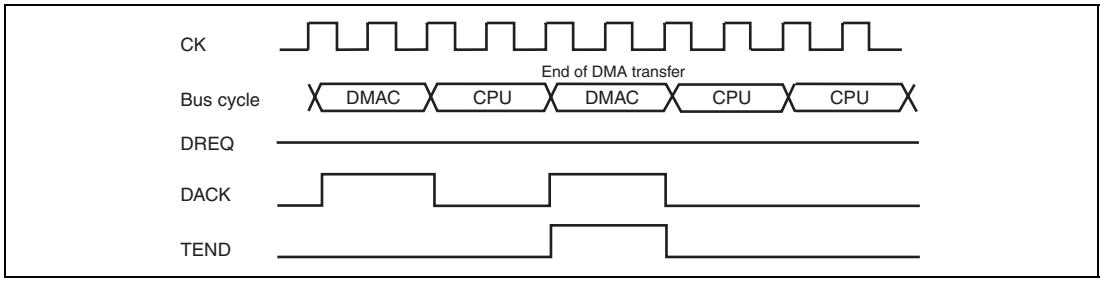
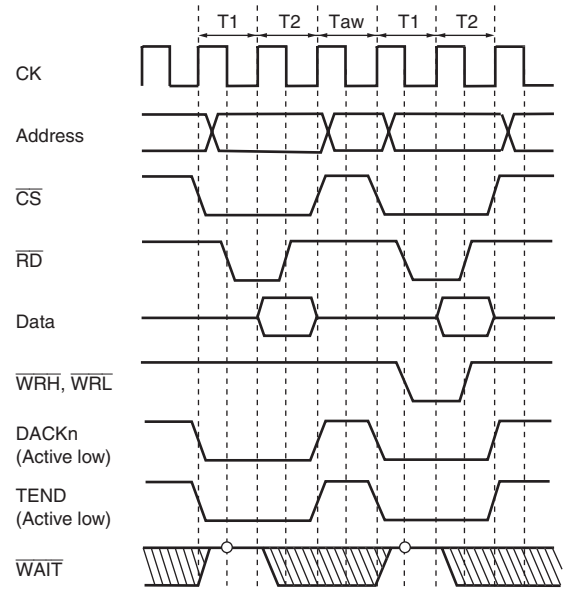


Figure 10.17 Example of DMA Transfer End Signal Timing (Cycle Steal Mode Level Detection)

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 (TEI) 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 (TEI) 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 (TEI_0)	IE	TE	High
	Data transfer half end interrupt (HEI_0)	HIE	HE	
1	Data transfer end interrupt (TEI_1)	IE	TE	
	Data transfer half end interrupt (HEI_1)	HIE	HE	
2	Data transfer end interrupt (TEI_2)	IE	TE	
	Data transfer half end interrupt (HEI_2)	HIE	HE	
3	Data transfer end interrupt (TEI_3)	IE	TE	
	Data transfer half end interrupt (HEI_3)	HIE	HE	
4	Data transfer end interrupt (TEI_4)	IE	TE	
	Data transfer half end interrupt (HEI_4)	HIE	HE	
5	Data transfer end interrupt (TEI_5)	IE	TE	
	Data transfer half end interrupt (HEI_5)	HIE	HE	
6	Data transfer end interrupt (TEI_6)	IE	TE	
	Data transfer half end interrupt (HEI_6)	HIE	HE	
7	Data transfer end interrupt (TEI_7)	IE	TE	
	Data transfer half end interrupt (HEI_7)	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) (SH7239A and SH7237A only).

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.

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.

The MTU2 operating frequency differs depending on whether the MTU2 is used for the complementary PWM mode output function or other functions as follows:

- Operating frequency for complementary PWM mode output function
  A maximum of 100 MHz: SH7239B and SH7237B
  A maximum of 80 MHz: SH7239A and SH7237A
- Operating frequency for other functions
  A maximum of 50 MHz: SH7239B and SH7237B
  A maximum of 40 MHz: SH7239A and SH7237A

Table 11.1 MTU2 Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 TCLKA TCLKB TCLKC TCLKD	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 TCLKA TCLKB	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/1024 TCLKA TCLKB TCLKC	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 Mϕ/1024 TCLKA TCLKB	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/64 Mϕ/256 Mϕ/1024 TCLKA TCLKB	Mϕ/1 Mϕ/4 Mϕ/16 Mϕ/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 1A	• Compare match or input capture 2A	• Compare match or input capture 3A	• Compare match or input capture 4A	• Compare match or input capture 5U
	• Compare match or input capture 0B	• Compare match or input capture 1B	• Compare match or input capture 2B	• Compare match or input capture 3B	• Compare match or input capture 4B	• Compare match or input capture 5V
	• Compare match or input capture 0C	• Overflow • Underflow	• Overflow • Underflow	• Compare match or input capture 3C	• Compare match or input capture 4C	• Compare match or input capture 5W
	• Compare match or input capture 0D			• Compare match or input capture 3D	• Compare match or input capture 4D	
	• Compare match 0E			• Overflow	• Overflow or underflow	
	• Compare match 0F					
	• Overflow					

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 TADCOR A_4 and TCNT_4 • A/D converter start request at a match between TADCOR B_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.

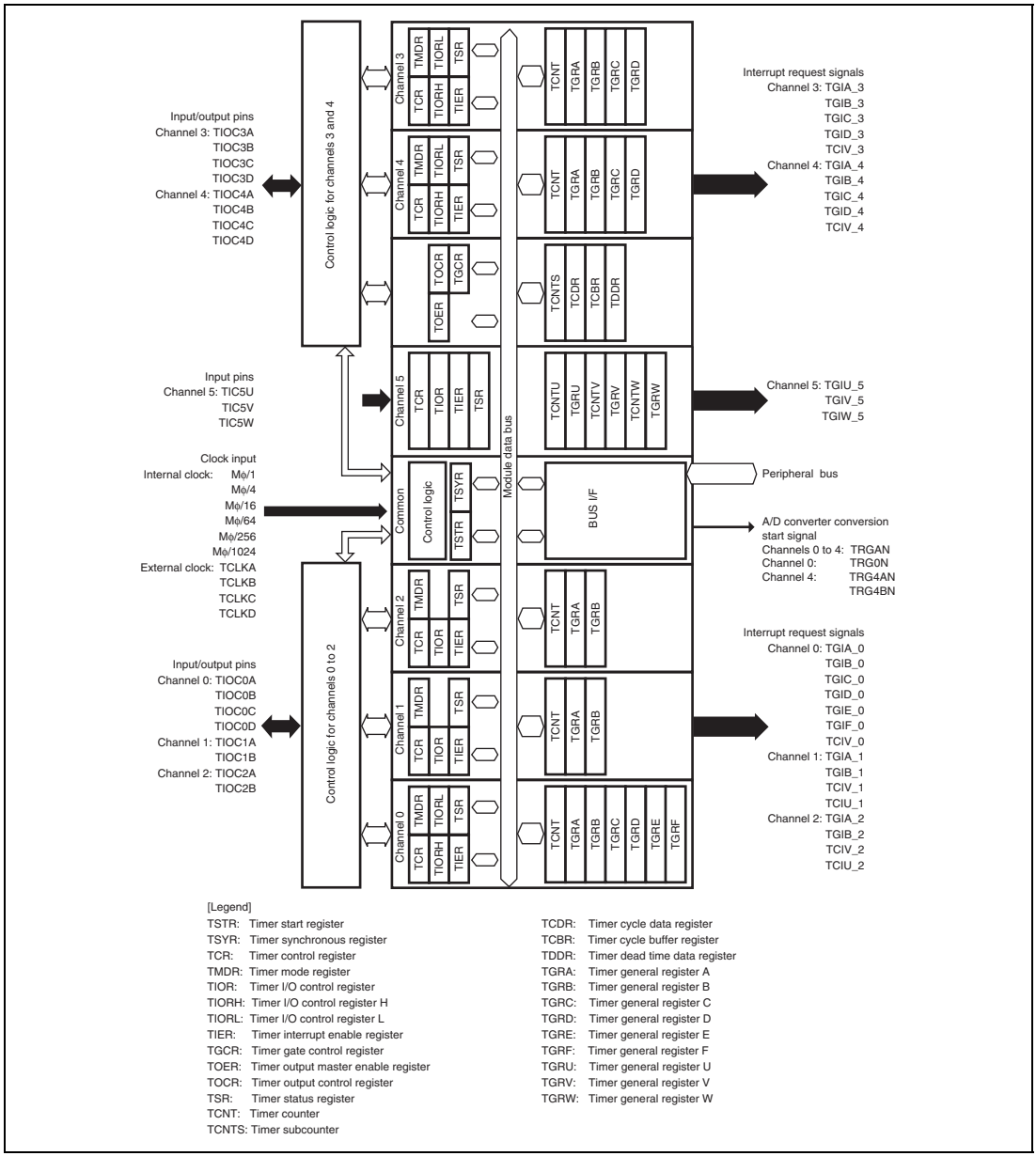


Figure 11.1 Block Diagram of MTU2

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 28, 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:	7	6	5	4	3	2	1	0
	CCLR[2:0]			CKEG[1:0]		TPSC[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 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. Mϕ/4 both edges = Mϕ/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 Mϕ/4 or slower. When Mϕ/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 Mϕ/1
			1	Internal clock: counts on Mϕ/4
		1	0	Internal clock: counts on Mϕ/16
			1	Internal clock: counts on Mϕ/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 Mϕ/1
			1	Internal clock: counts on Mϕ/4
		1	0	Internal clock: counts on Mϕ/16
			1	Internal clock: counts on Mϕ/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 Mϕ/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 Mϕ/1
			1	Internal clock: counts on Mϕ/4
		1	0	Internal clock: counts on Mϕ/16
			1	Internal clock: counts on Mϕ/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 Mϕ/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 Mϕ/1
			1	Internal clock: counts on Mϕ/4
		1	0	Internal clock: counts on Mϕ/16
			1	Internal clock: counts on mϕ/64
	1	0	0	Internal clock: counts on Mϕ/256
			1	Internal clock: counts on Mϕ/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 Mϕ/1
		1	Internal clock: counts on Mϕ/4
	1	0	Internal clock: counts on Mϕ/16
		1	Internal clock: counts on Mϕ/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.

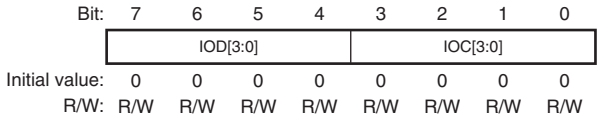
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- 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

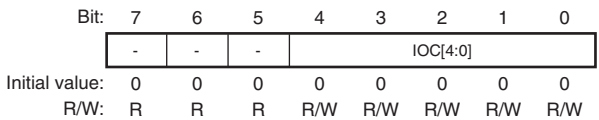
Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 11.12 TIOR_1: Table 11.14 TIOR_2: Table 11.15 TIORH_3: Table 11.16 TIORH_4: Table 11.18
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify 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	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	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 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_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
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

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] 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] 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.

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.

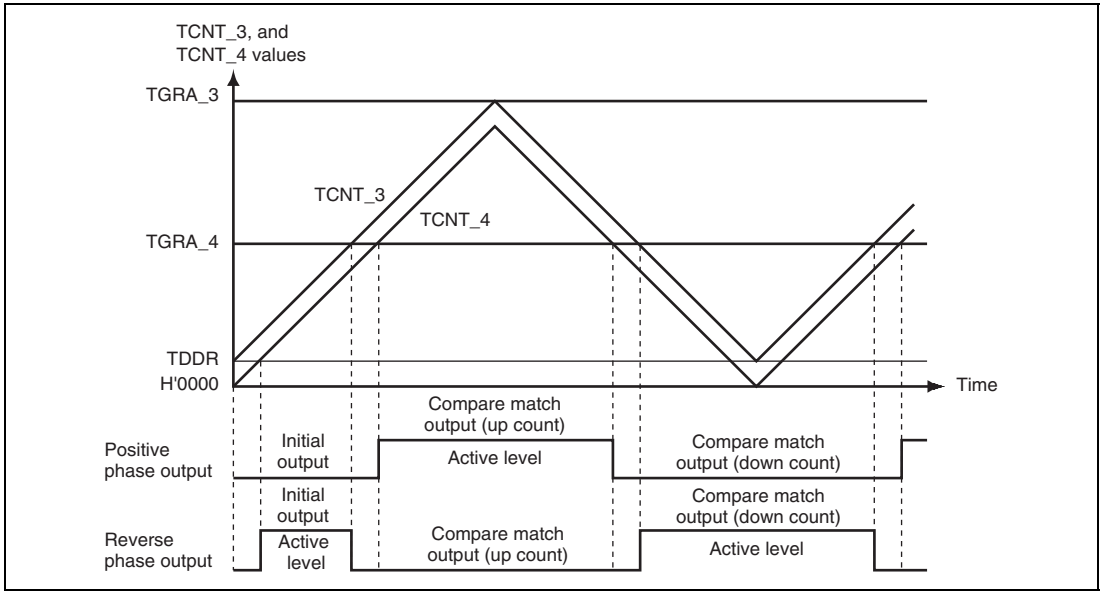


Figure 11.2 Complementary PWM Mode Output Level Example

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* 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* 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* 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* 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* 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*
				This bit selects the output level on TIOC3B in reset-synchronized PWM mode/complementary PWM mode. See table 11.38.

Note: * Setting the TOCS bit in TOCR1 to 1 makes this bit setting 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.

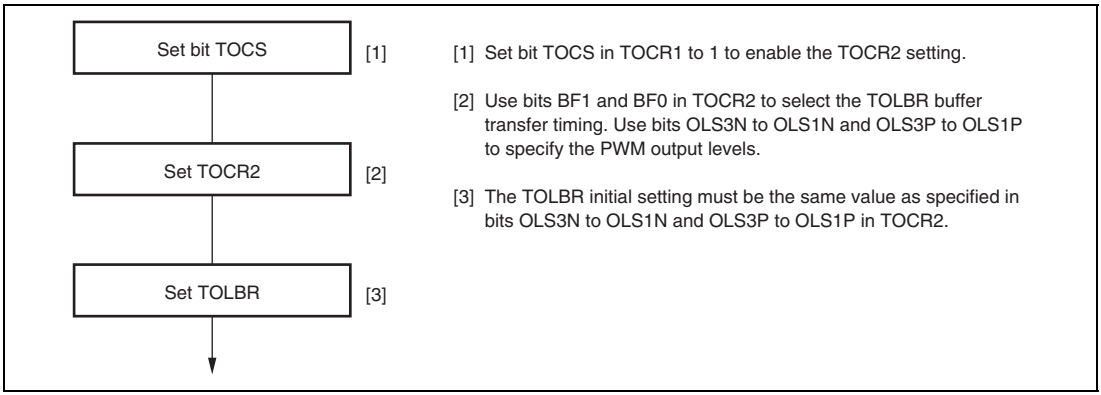


Figure 11.3 PWM Output Level Setting Procedure in Buffer Operation

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 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.

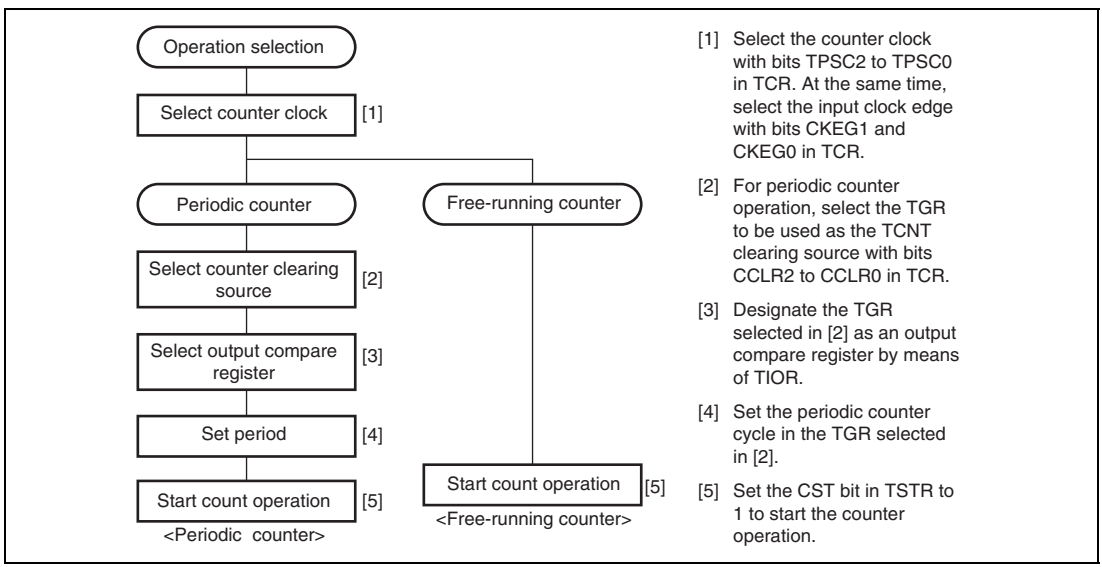


Figure 11.4 Example of Counter Operation Setting Procedure

(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.

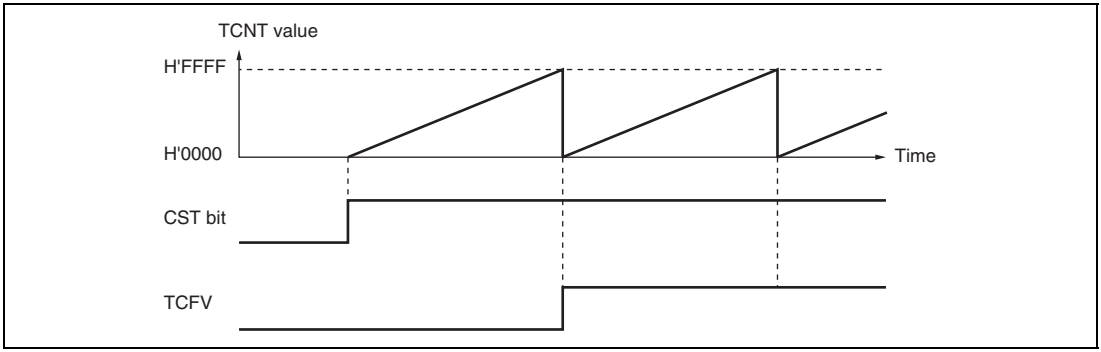


Figure 11.5 Free-Running Counter Operation

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.

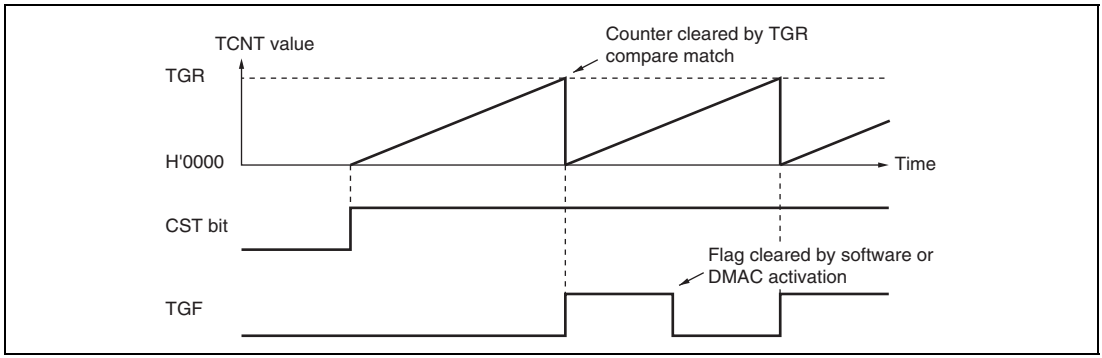


Figure 11.6 Periodic Counter Operation

(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.

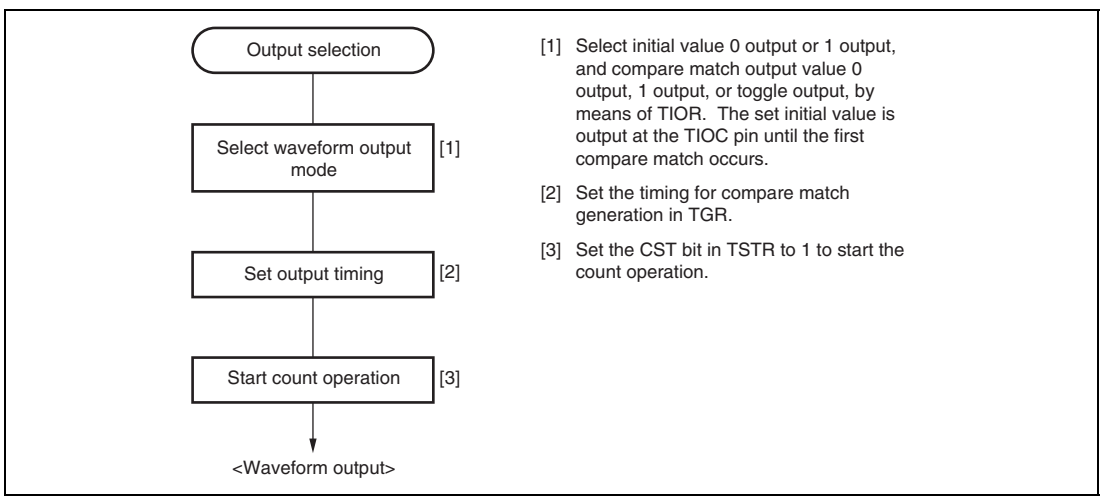


Figure 11.7 Example of Setting Procedure for Waveform Output by Compare Match

(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.

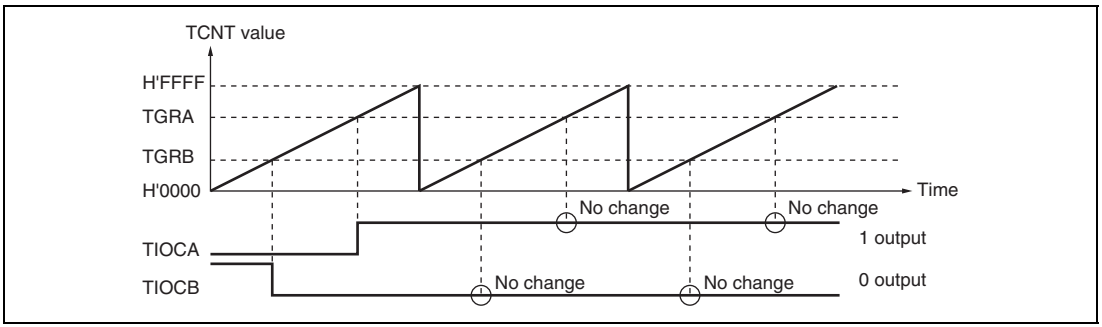


Figure 11.8 Example of 0 Output/1 Output Operation

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.

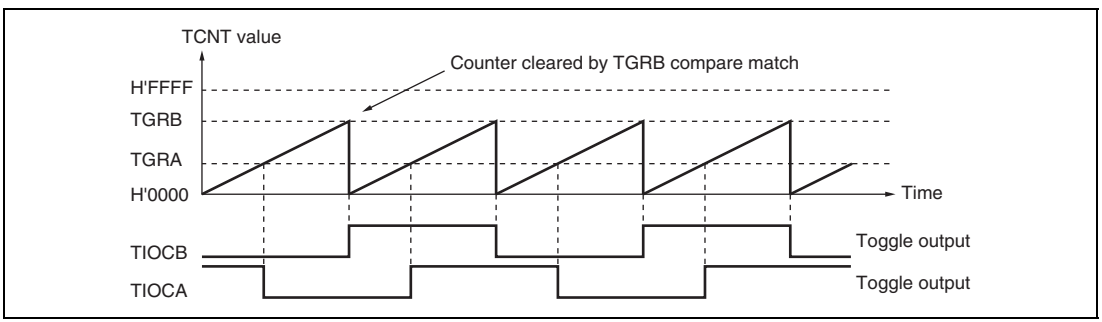


Figure 11.9 Example of Toggle Output Operation

(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, Mϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Mϕ/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 11.10 shows an example of the input capture operation setting procedure.

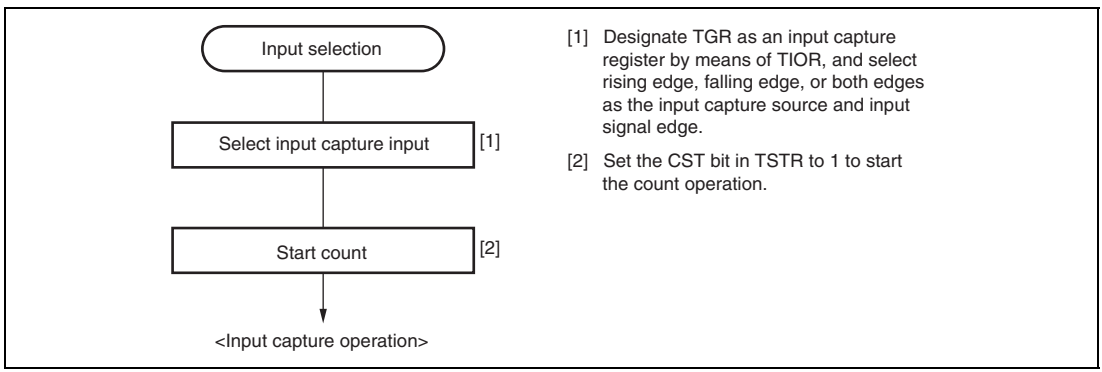


Figure 11.10 Example of Input Capture Operation Setting Procedure

(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.

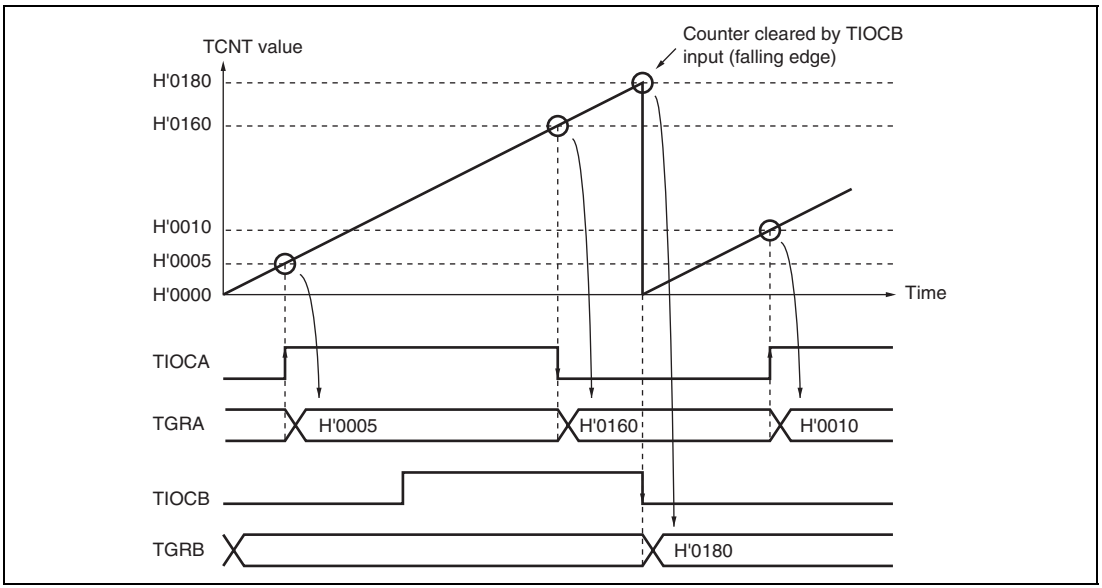


Figure 11.11 Example of Input Capture Operation

11.4.2 Synchronous Operation

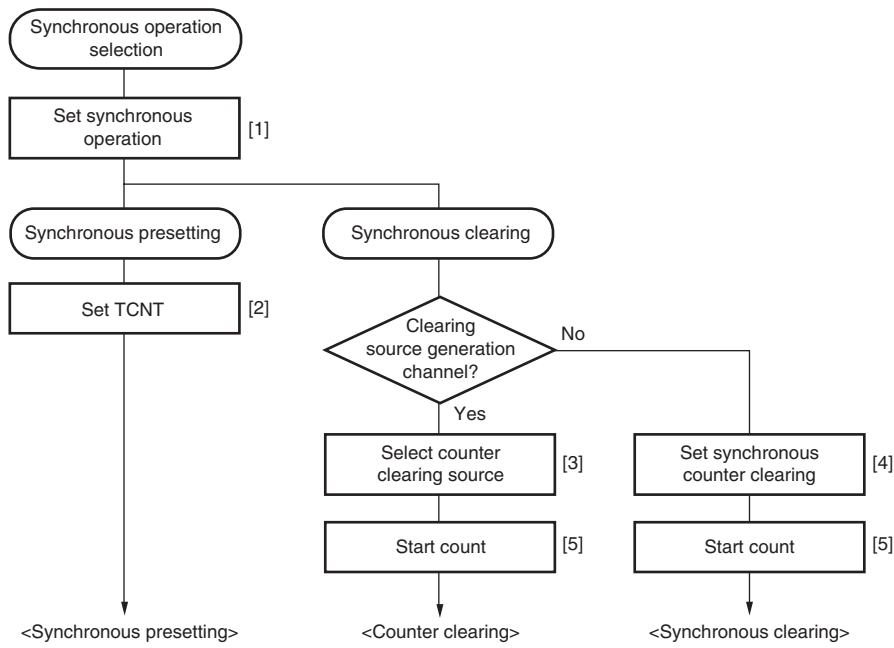
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

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.

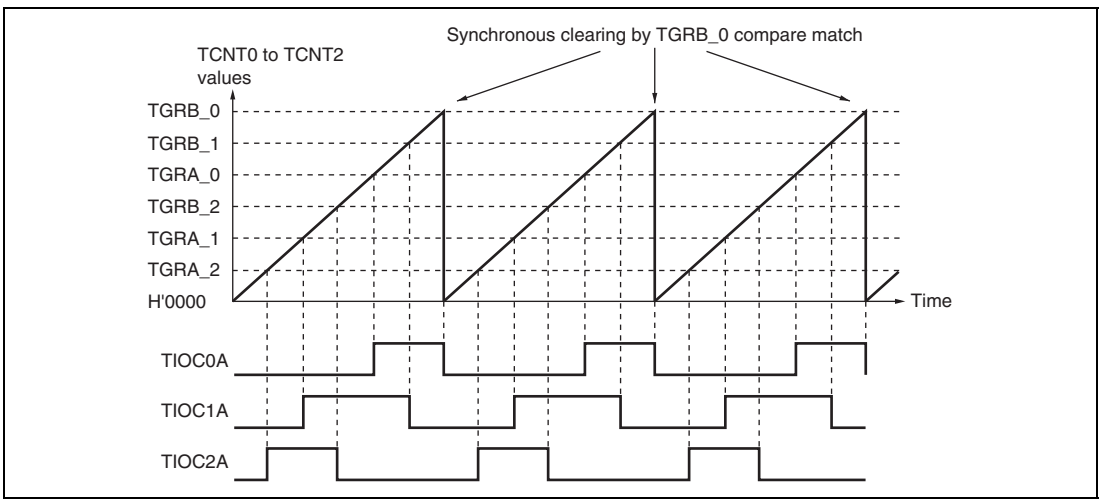


Figure 11.13 Example of Synchronous Operation

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.

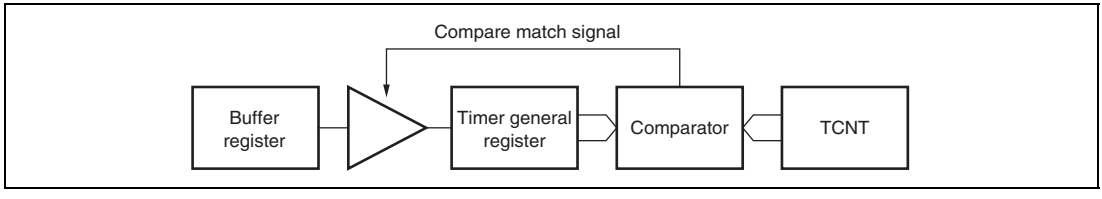


Figure 11.14 Compare Match Buffer Operation

- 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.

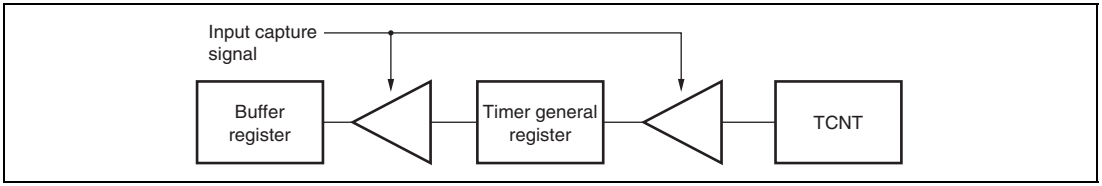


Figure 11.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 11.16 shows an example of the buffer operation setting procedure.

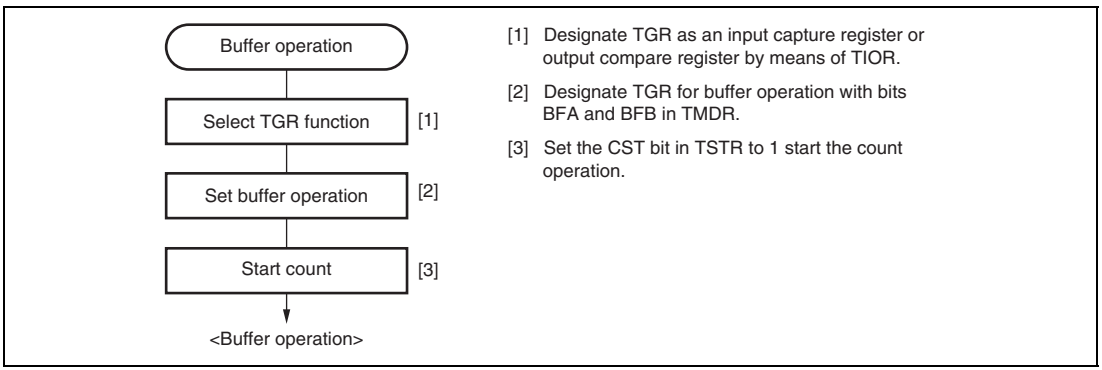


Figure 11.16 Example of Buffer Operation Setting Procedure

(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.

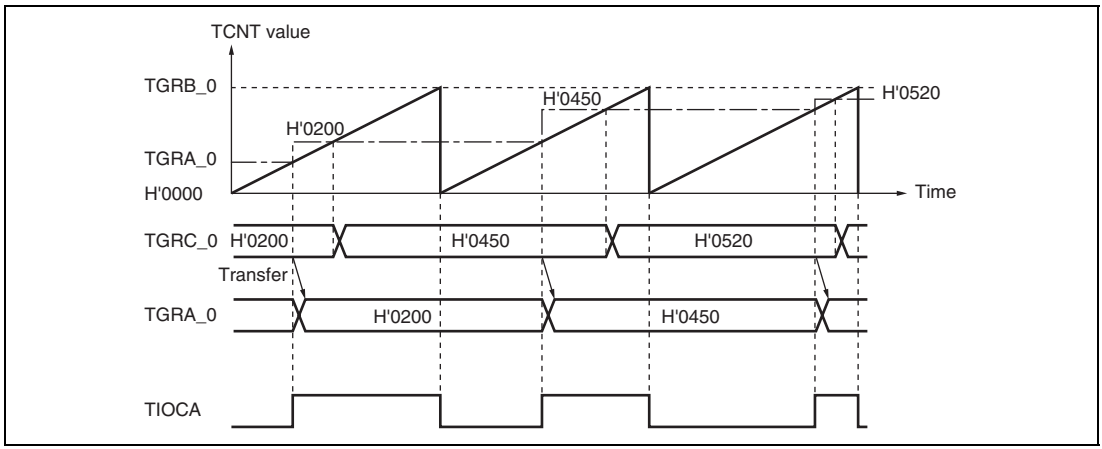


Figure 11.17 Example of Buffer Operation (1)

(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.

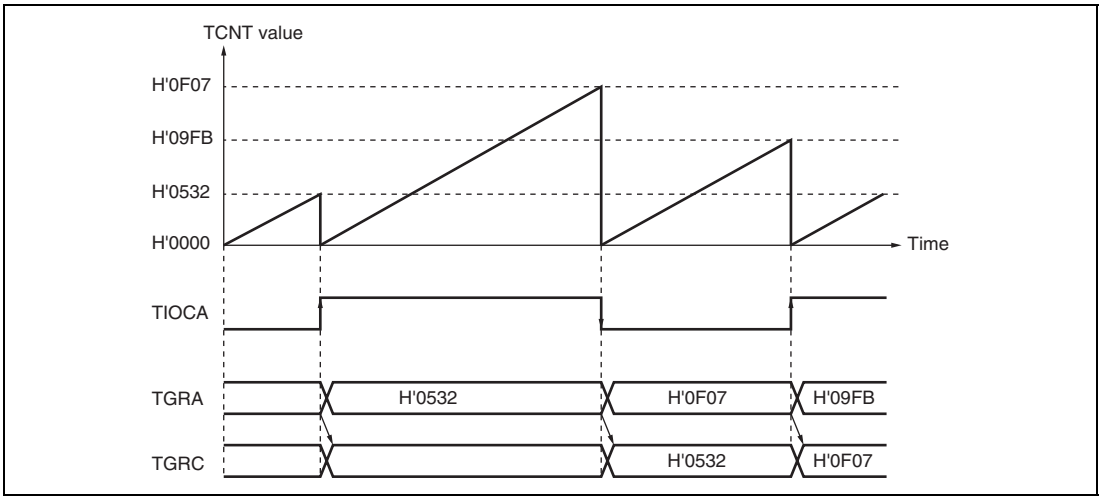


Figure 11.18 Example of Buffer Operation (2)

(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.

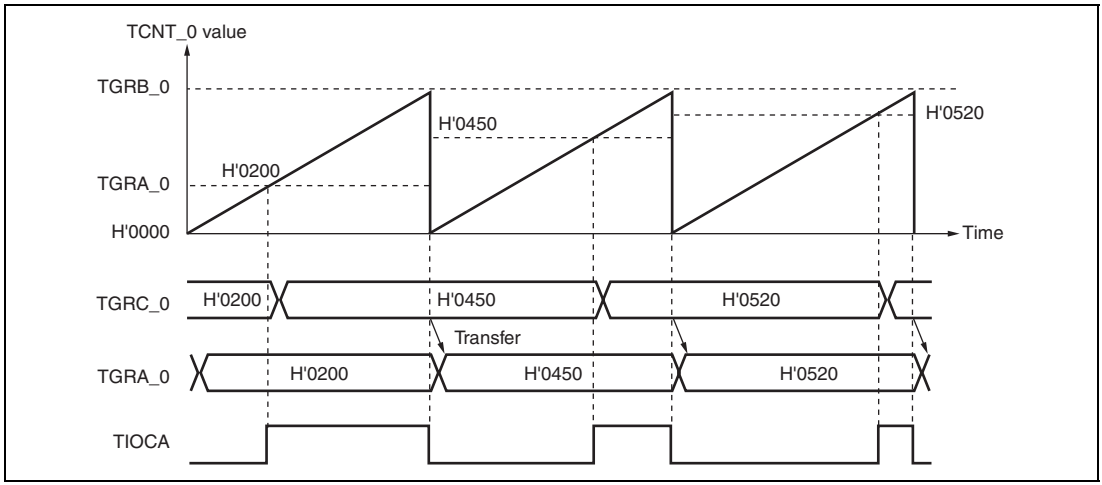


Figure 11.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

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). 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.

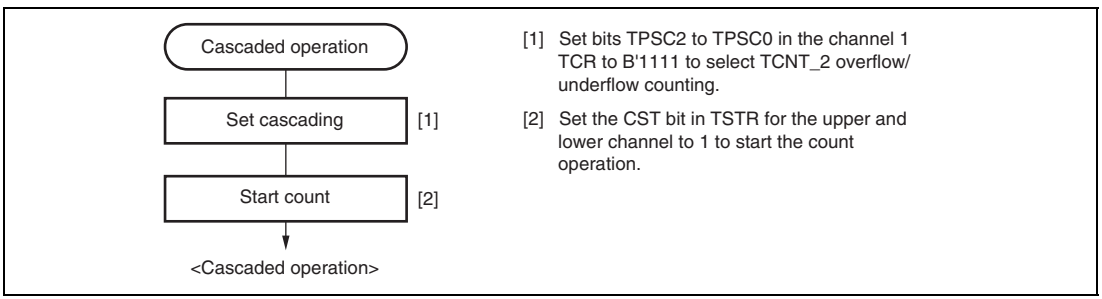


Figure 11.20 Cascaded Operation Setting Procedure

(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.

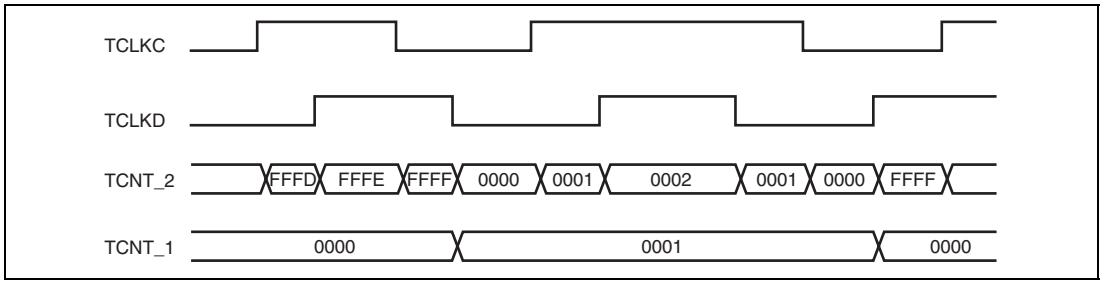


Figure 11.21 Cascaded Operation Example (a)

(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.

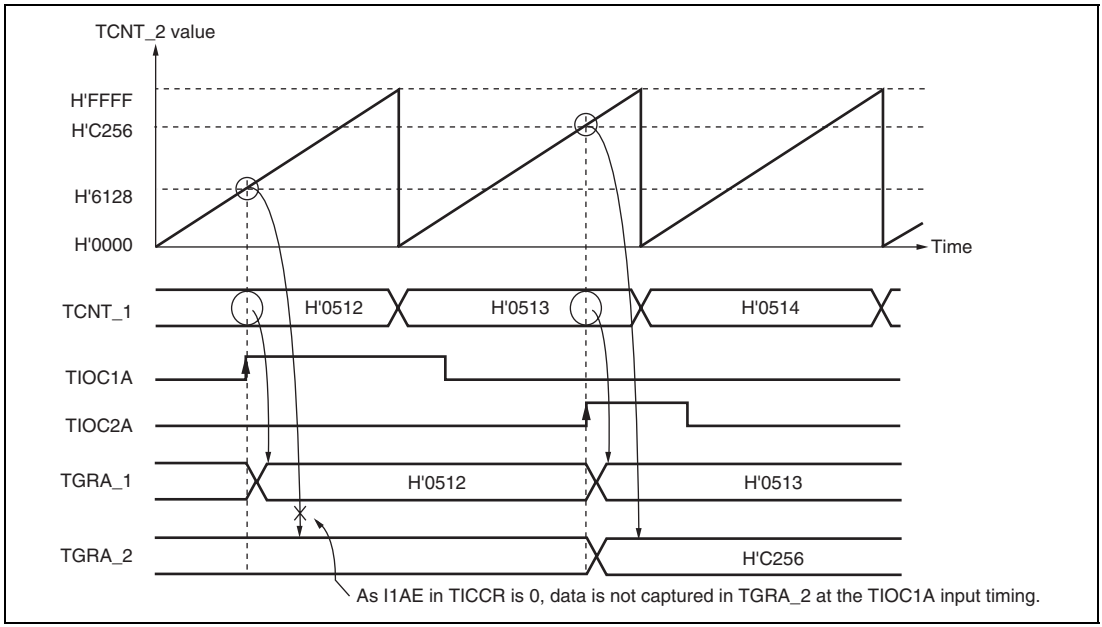


Figure 11.22 Cascaded Operation Example (b)

(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.

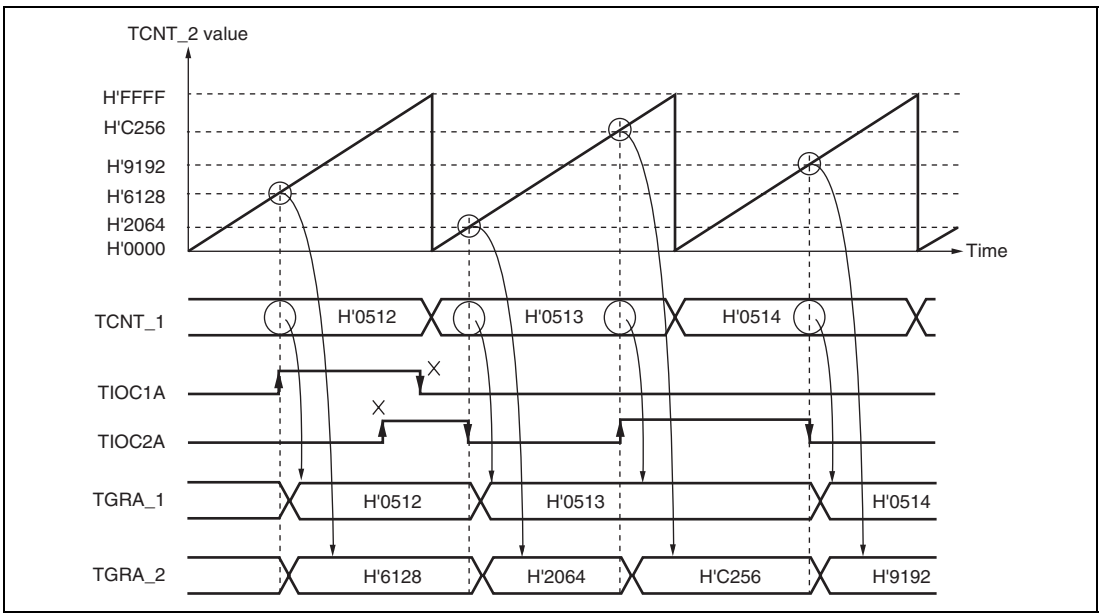


Figure 11.23 Cascaded Operation Example (c)

(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.

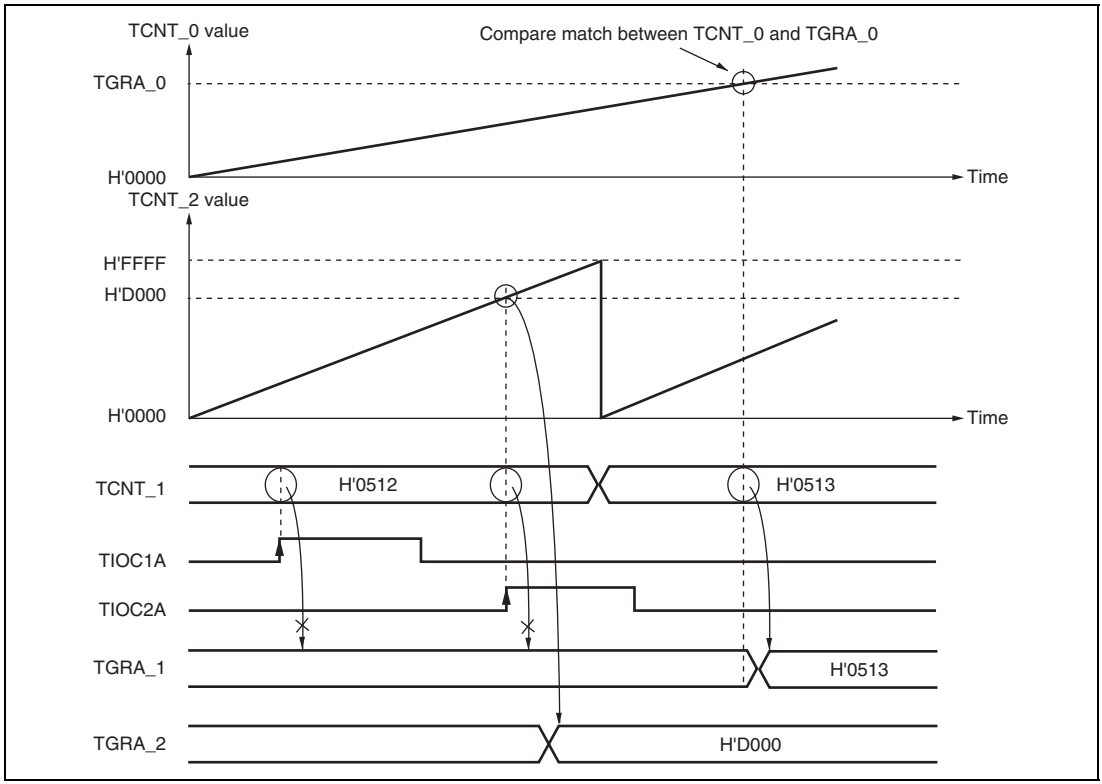


Figure 11.24 Cascaded Operation Example (d)

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 a synchronization 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.

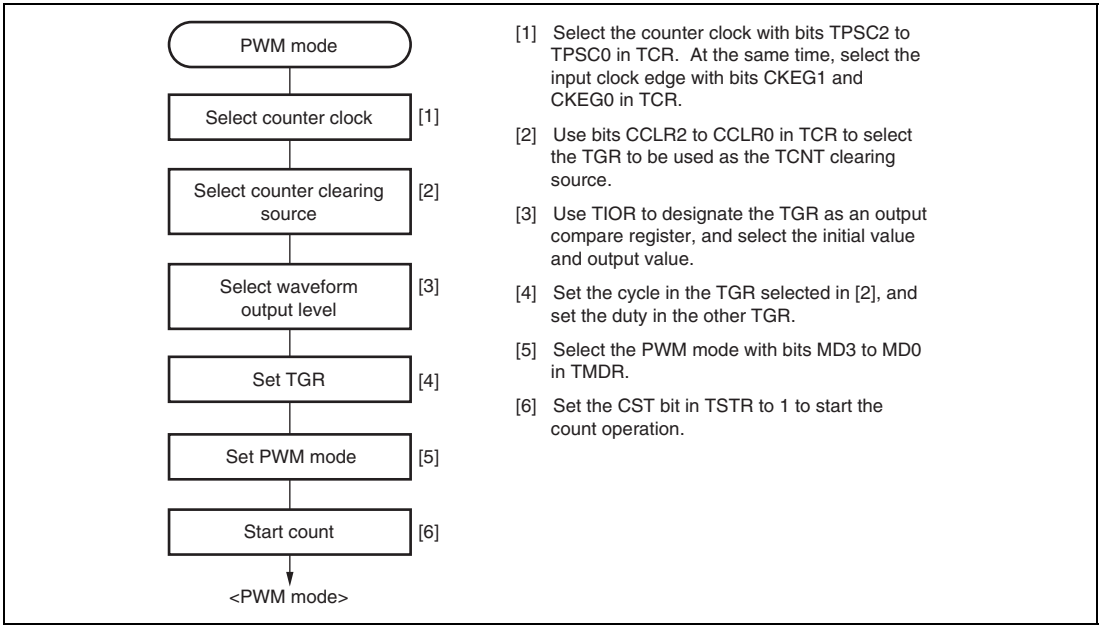


Figure 11.25 Example of PWM Mode Setting Procedure

(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.

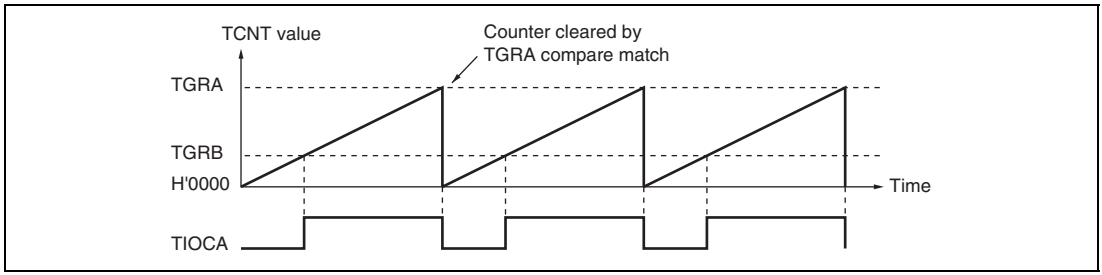


Figure 11.26 Example of PWM Mode Operation (1)

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.

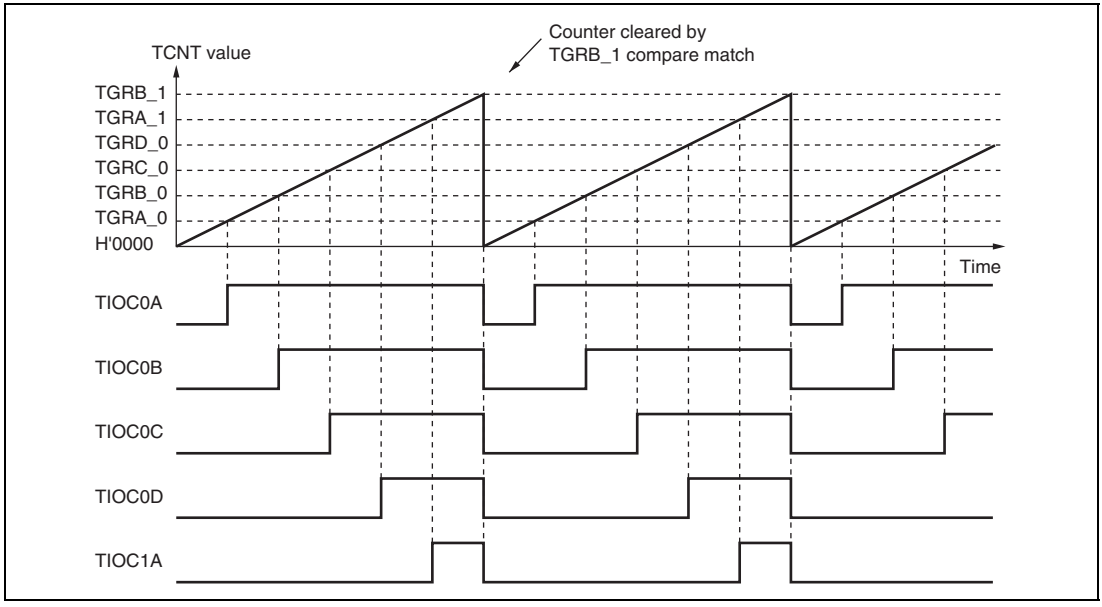


Figure 11.27 Example of PWM Mode Operation (2)

Figure 11.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

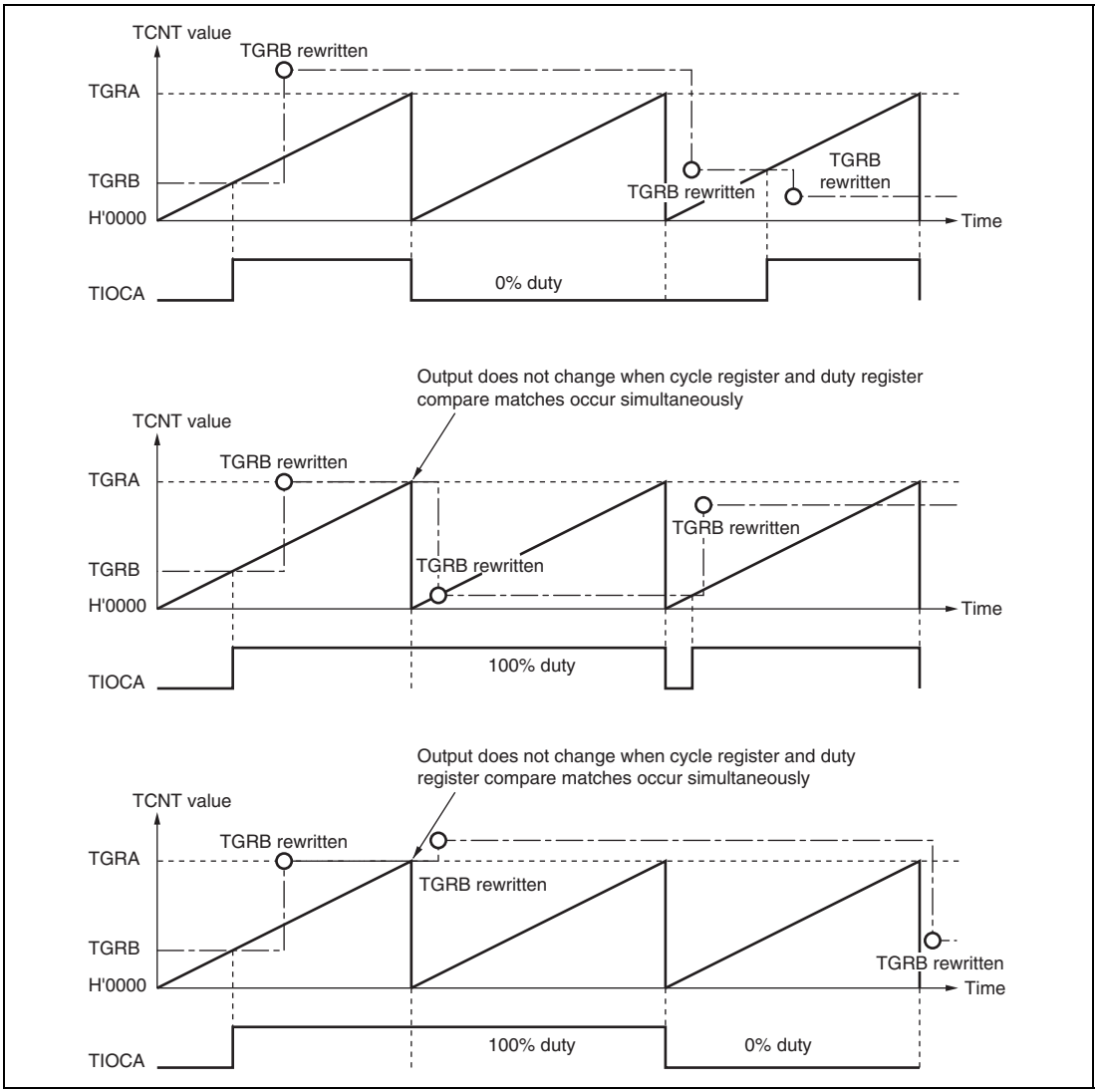


Figure 11.28 Example of PWM Mode Operation (3)

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.

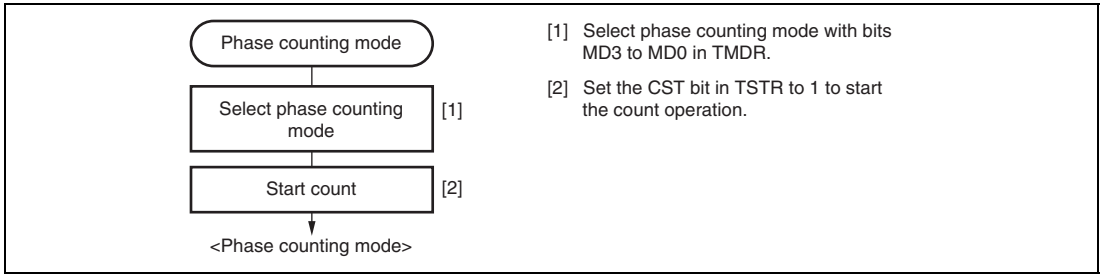


Figure 11.29 Example of Phase Counting Mode Setting Procedure

(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.

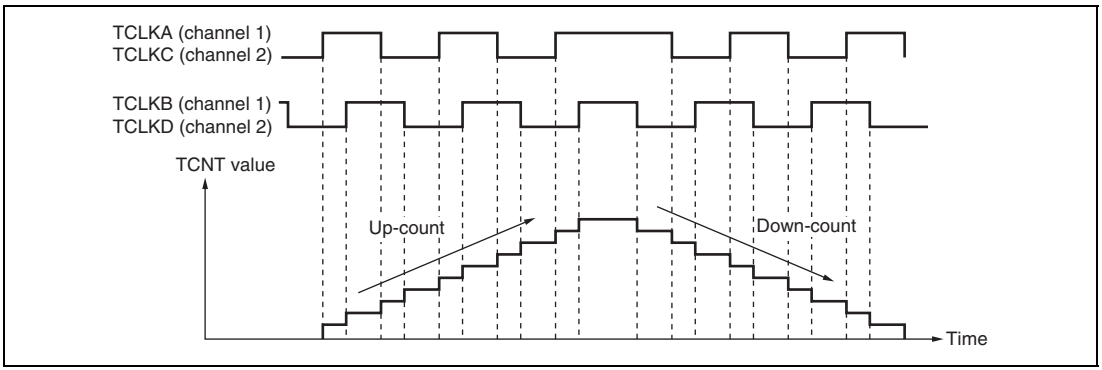


Figure 11.30 Example of Phase Counting Mode 1 Operation

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.

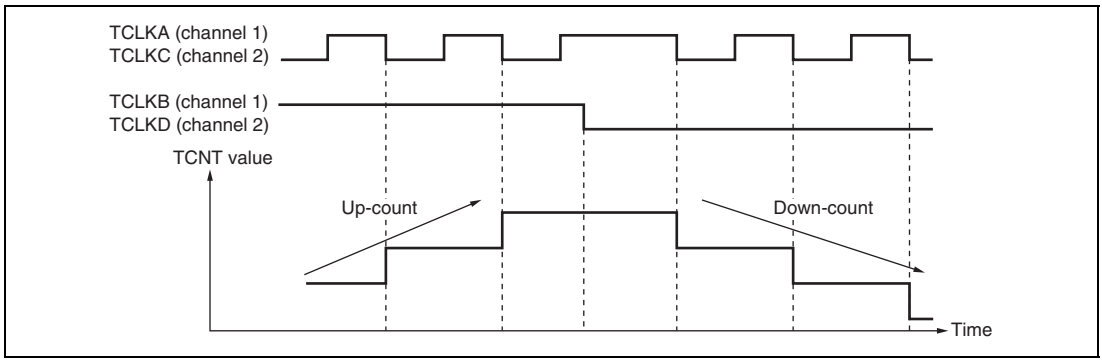


Figure 11.31 Example of Phase Counting Mode 2 Operation

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.

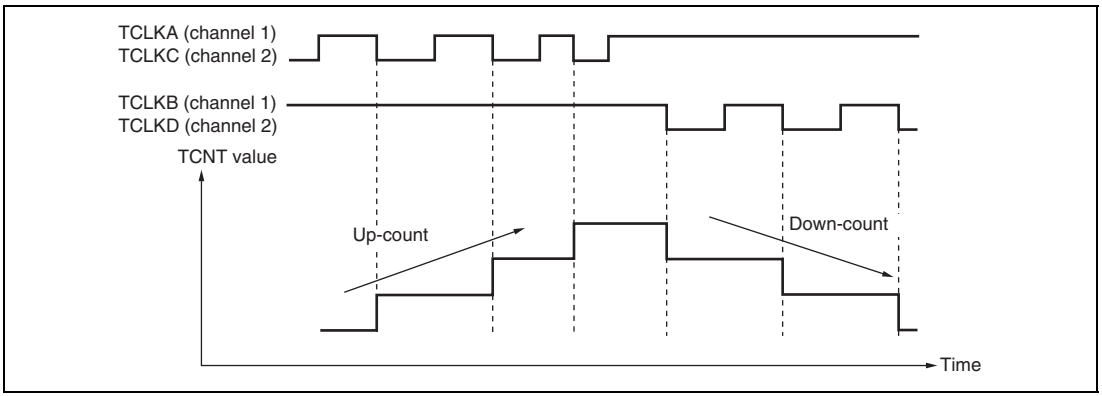


Figure 11.32 Example of Phase Counting Mode 3 Operation

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.

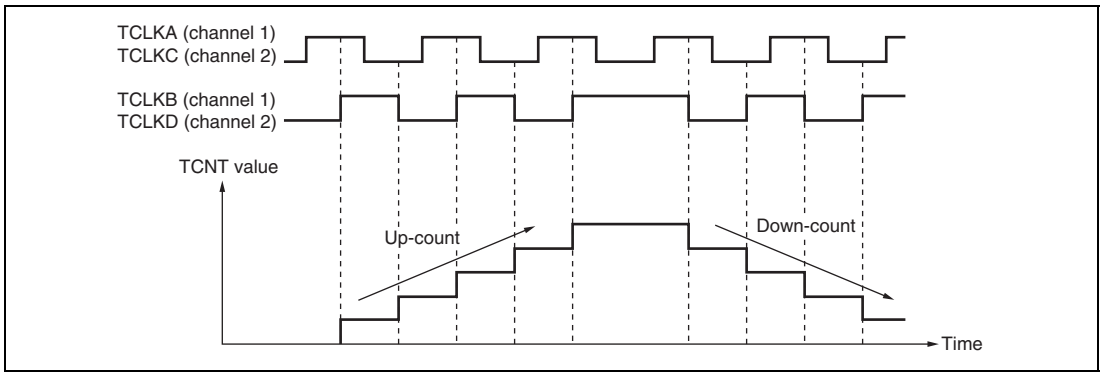


Figure 11.33 Example of Phase Counting Mode 4 Operation

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.

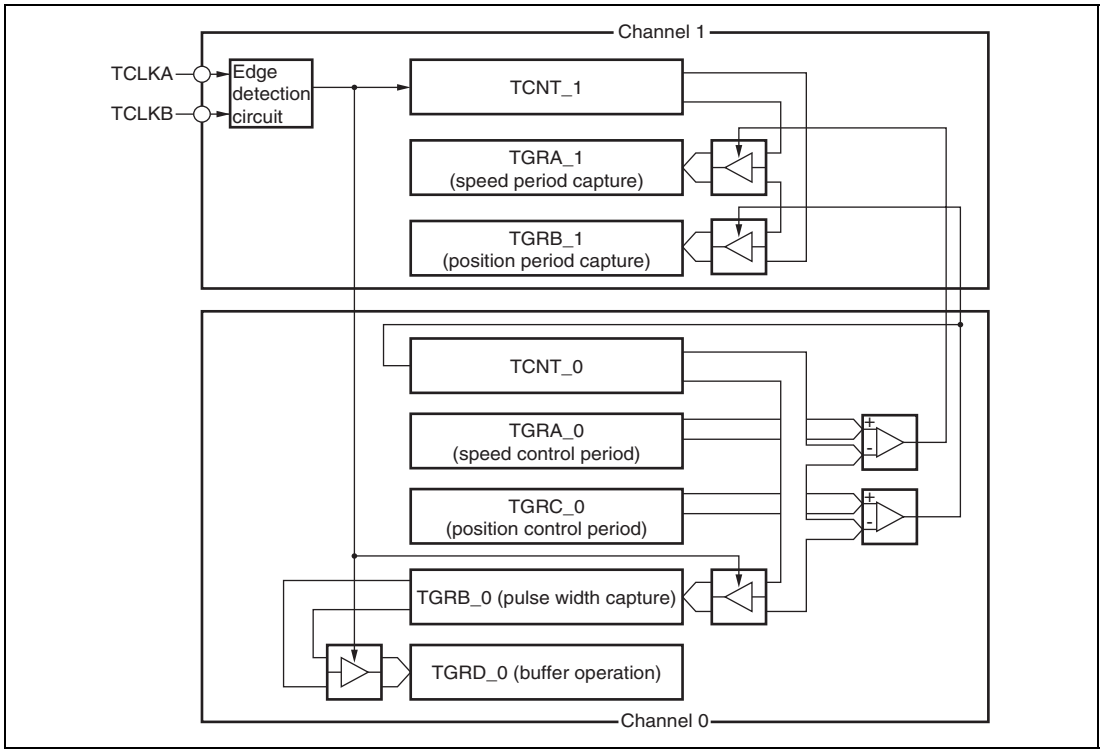


Figure 11.34 Phase Counting Mode Application Example

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.

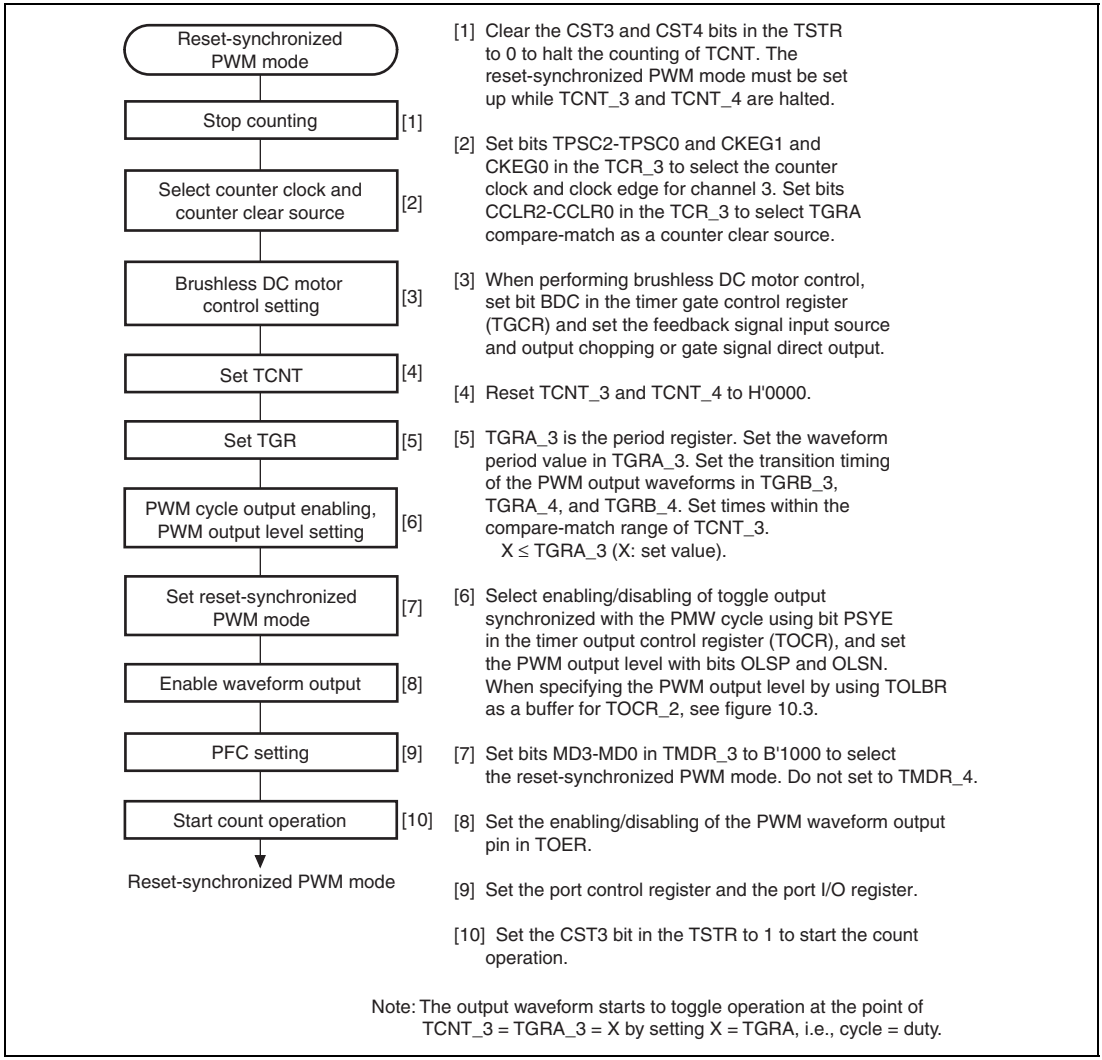


Figure 11.35 Procedure for Selecting Reset-Synchronized PWM Mode

(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.

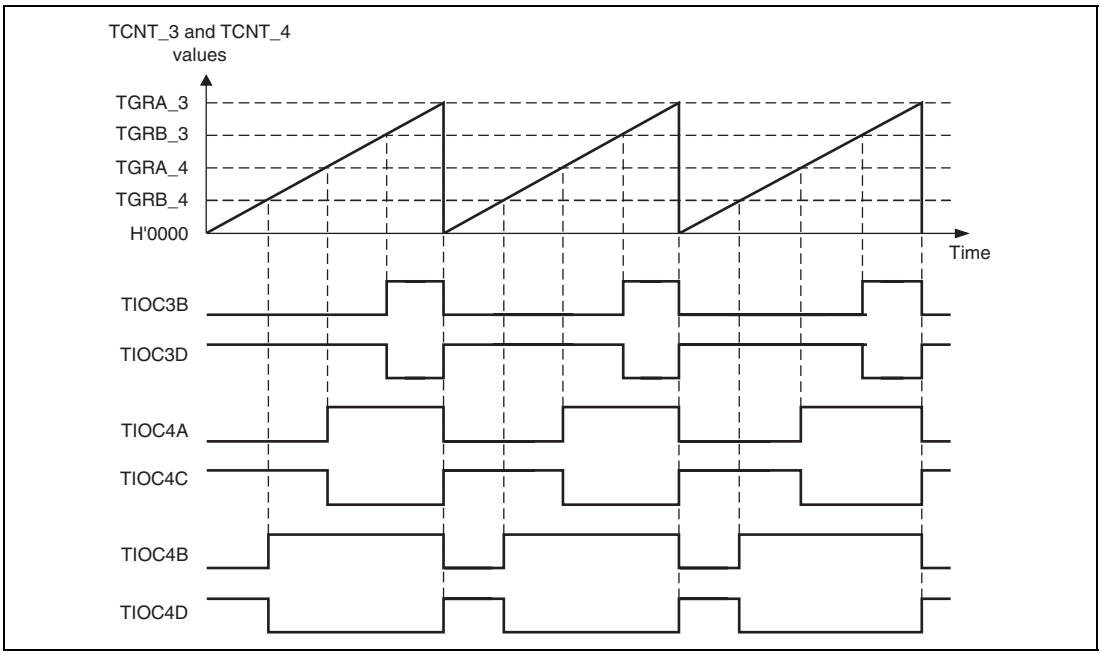


Figure 11.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

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).

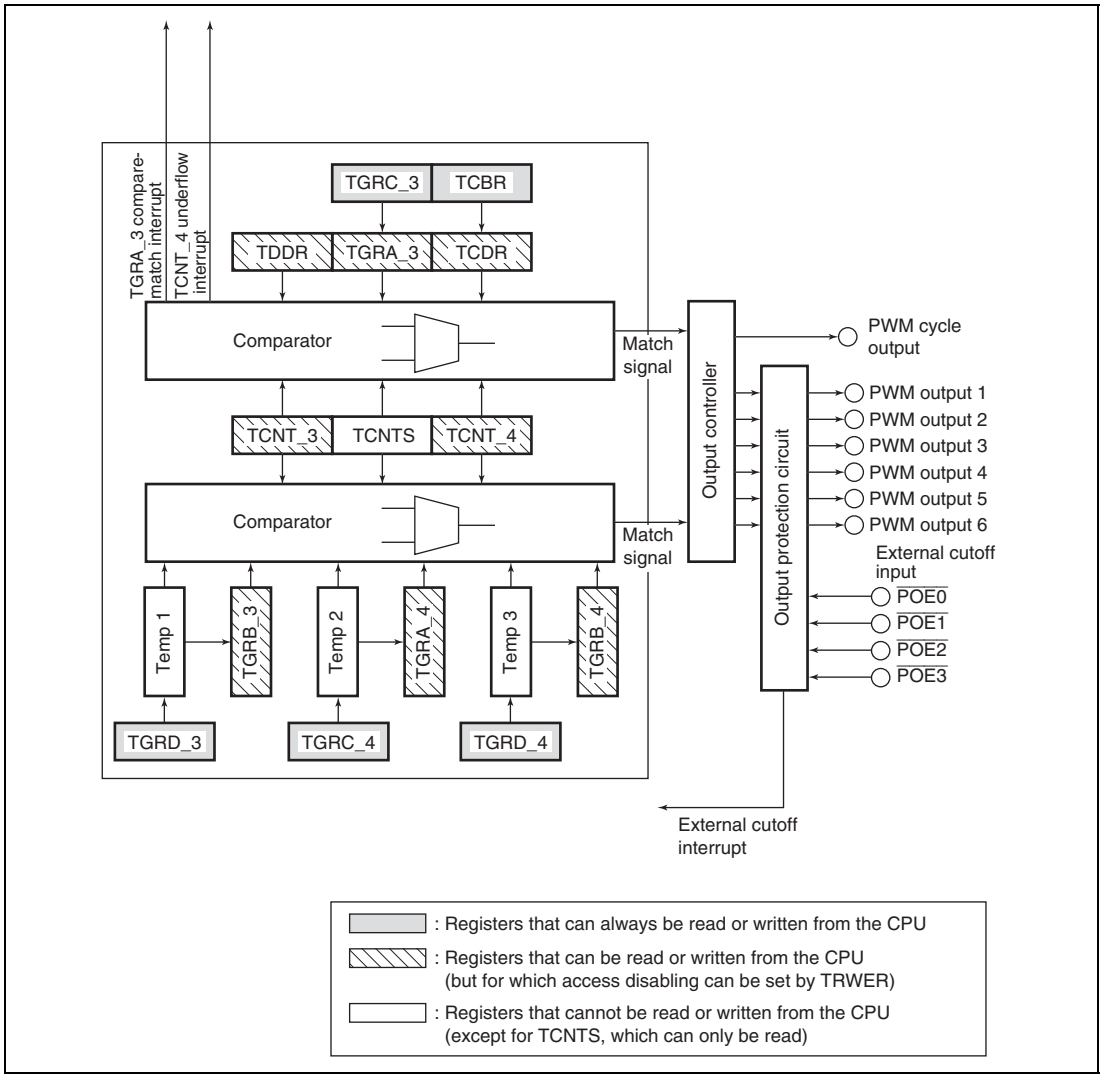
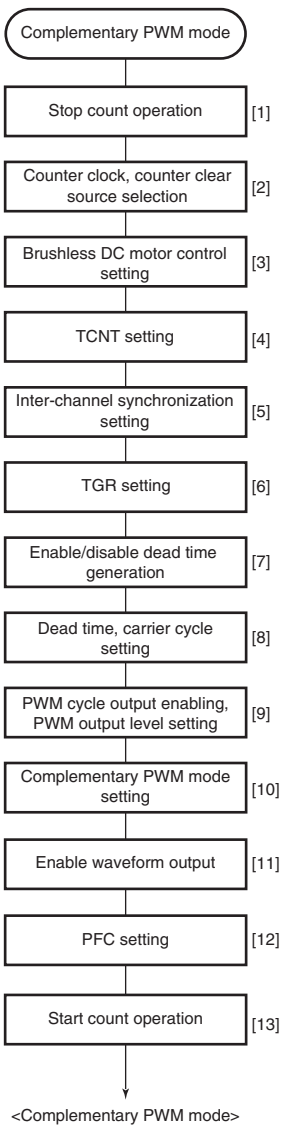


Figure 11.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(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 carrier cycle data register (TCDR) and carrier 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.

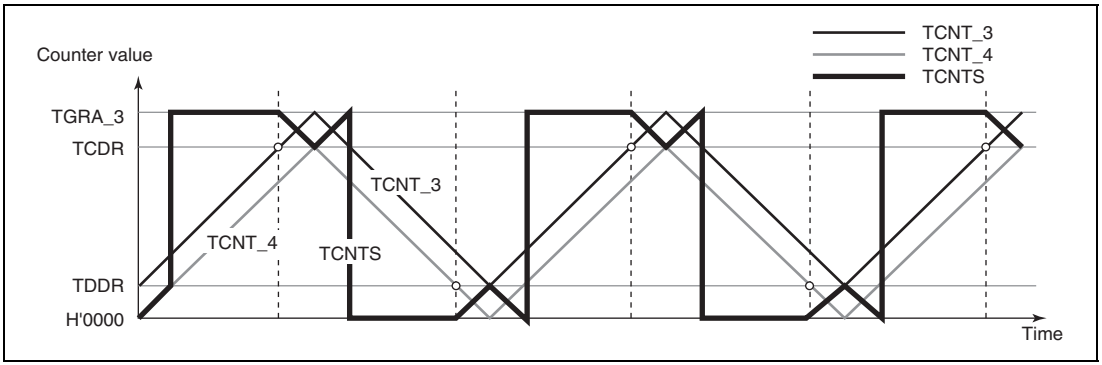


Figure 11.39 Complementary PWM Mode Counter Operation

(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.

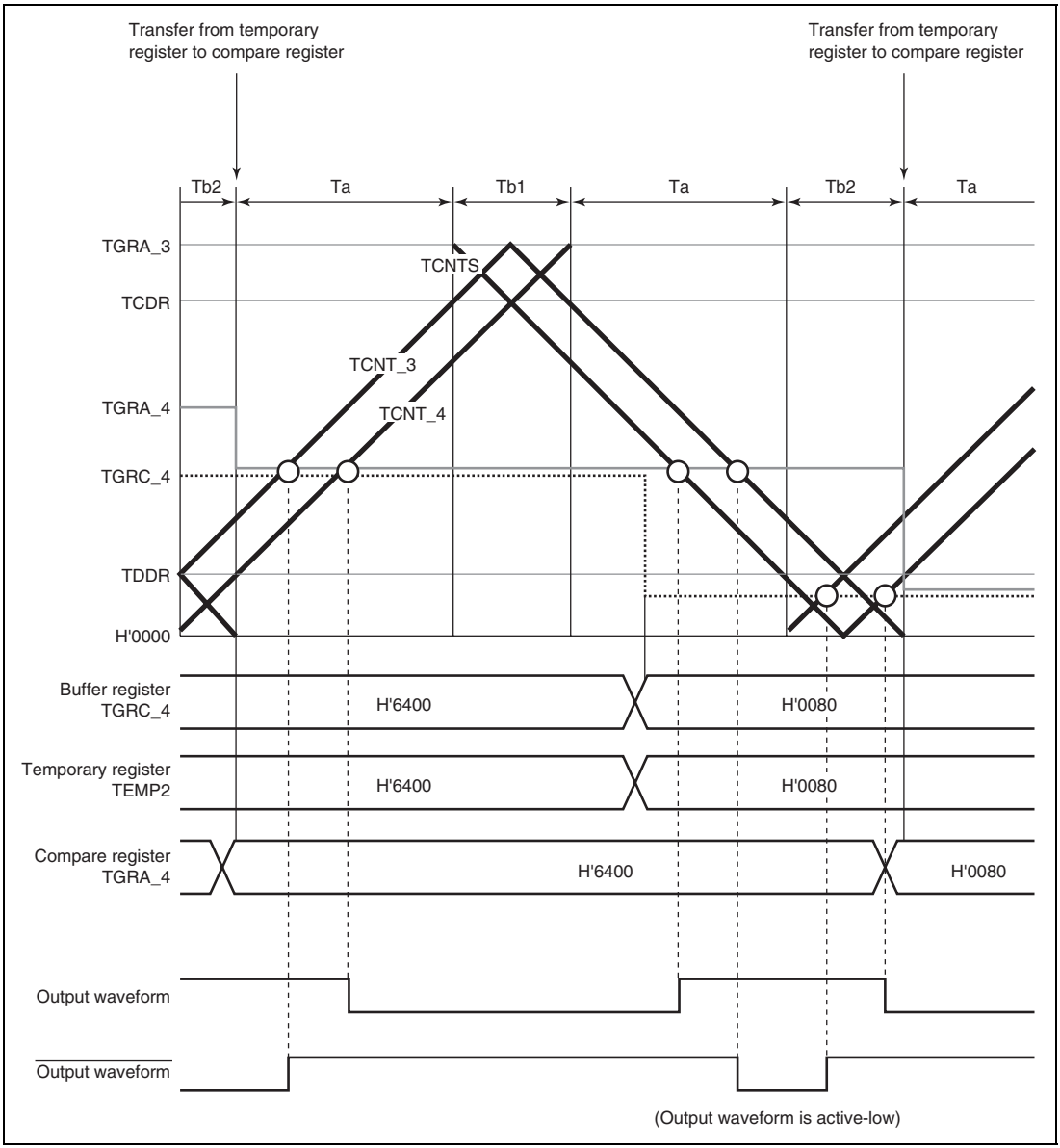


Figure 11.40 Example of Complementary PWM Mode Operation

(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.

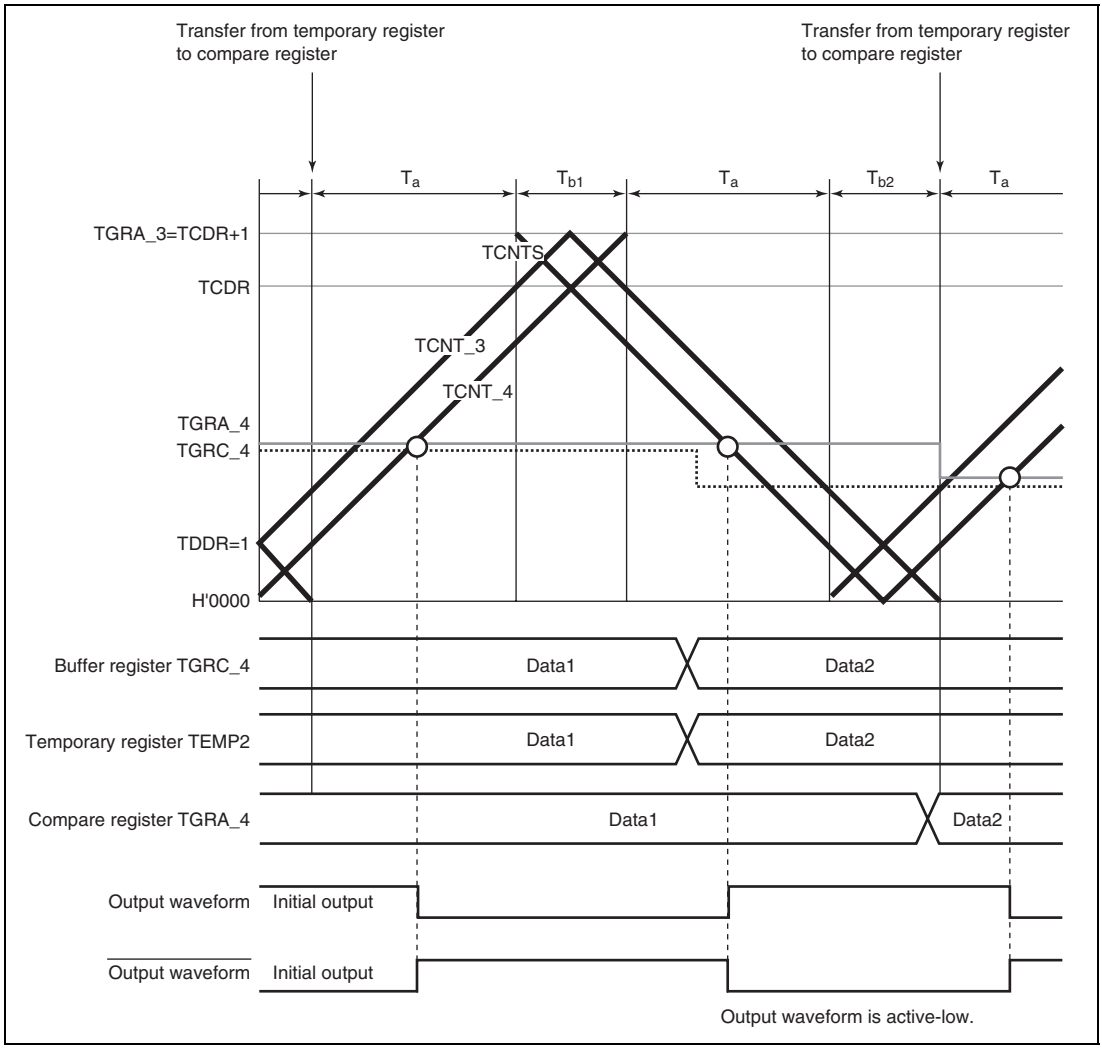


Figure 11.41 Example of Operation without Dead Time

(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
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.

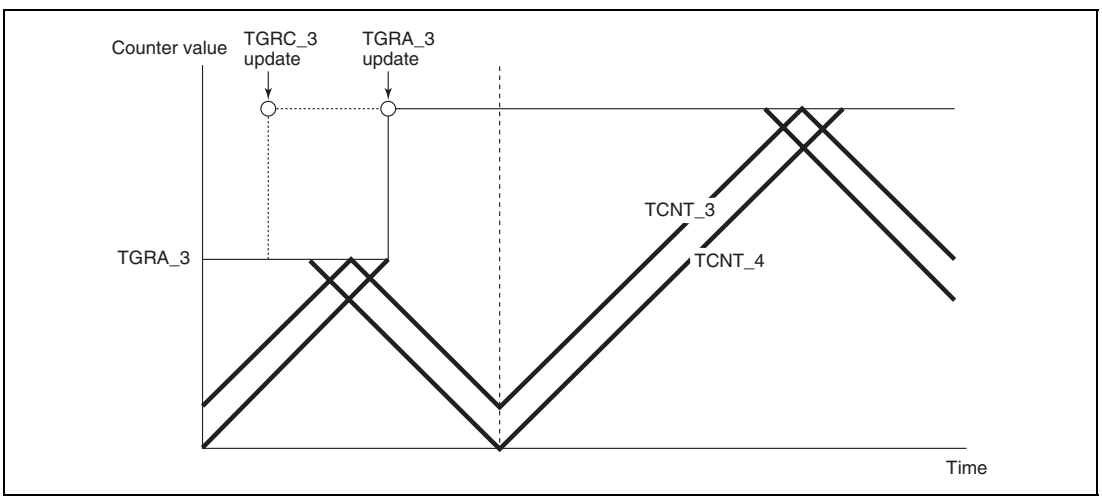


Figure 11.42 Example of PWM Cycle Updating

(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 in complementary PWM mode 3 (transfers at crests and troughs), 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.

When TCNT3 counts downward in complementary PWM mode 1 (transfers at crests), if buffer register is updated, transfer to the temporary register is not performed until counting-up is started. When TCNT3 counts upward in complementary PWM mode 2 (transfers at troughs), if buffer register is updated, transfer to the temporary register is not performed until counting-down is started.

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.

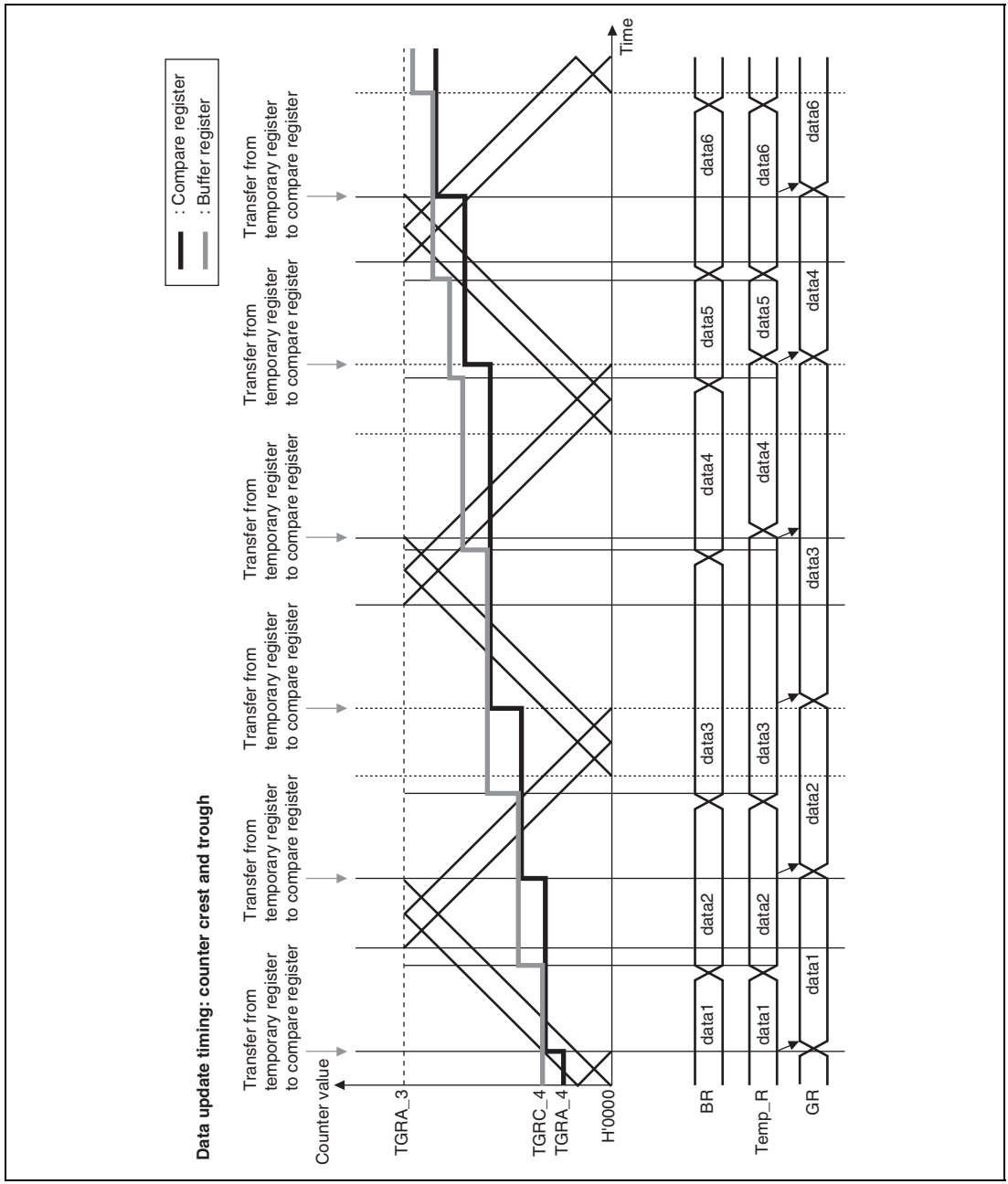


Figure 11.43 Example of Data Update in Complementary PWM Mode

(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.

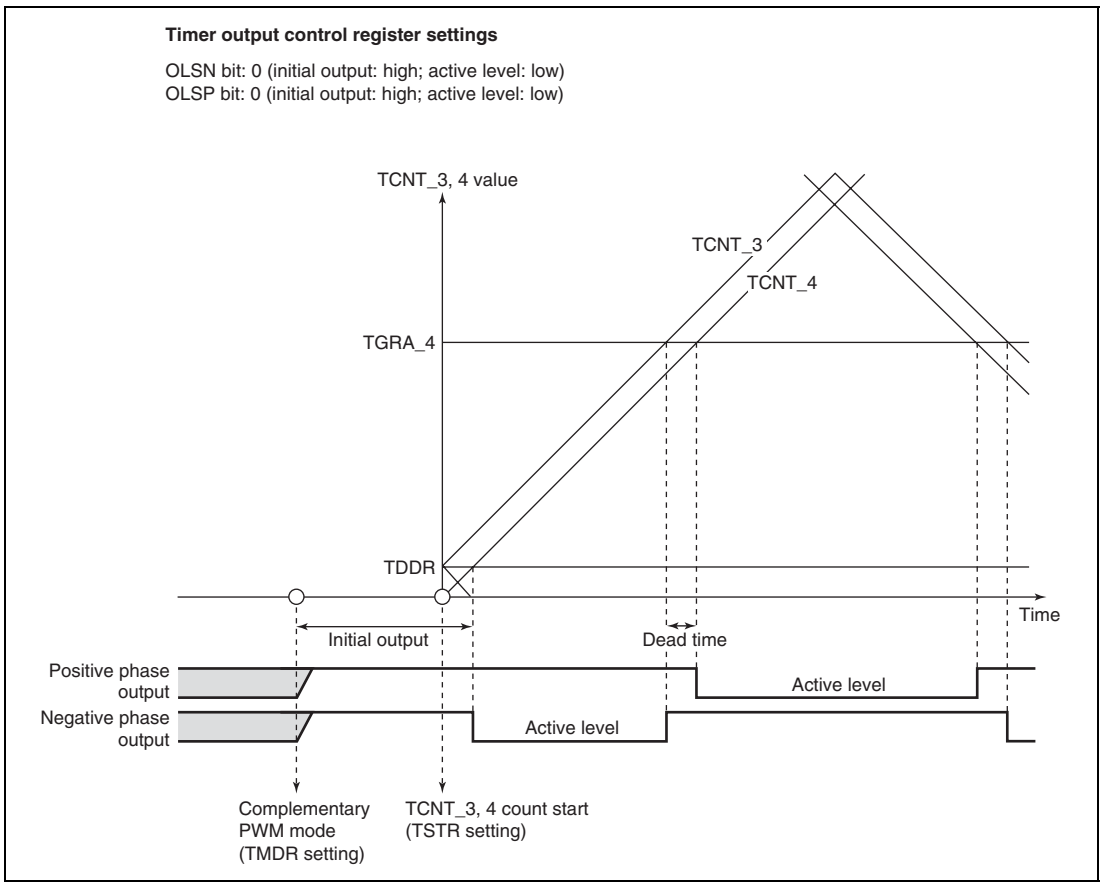


Figure 11.44 Example of Initial Output in Complementary PWM Mode (1)

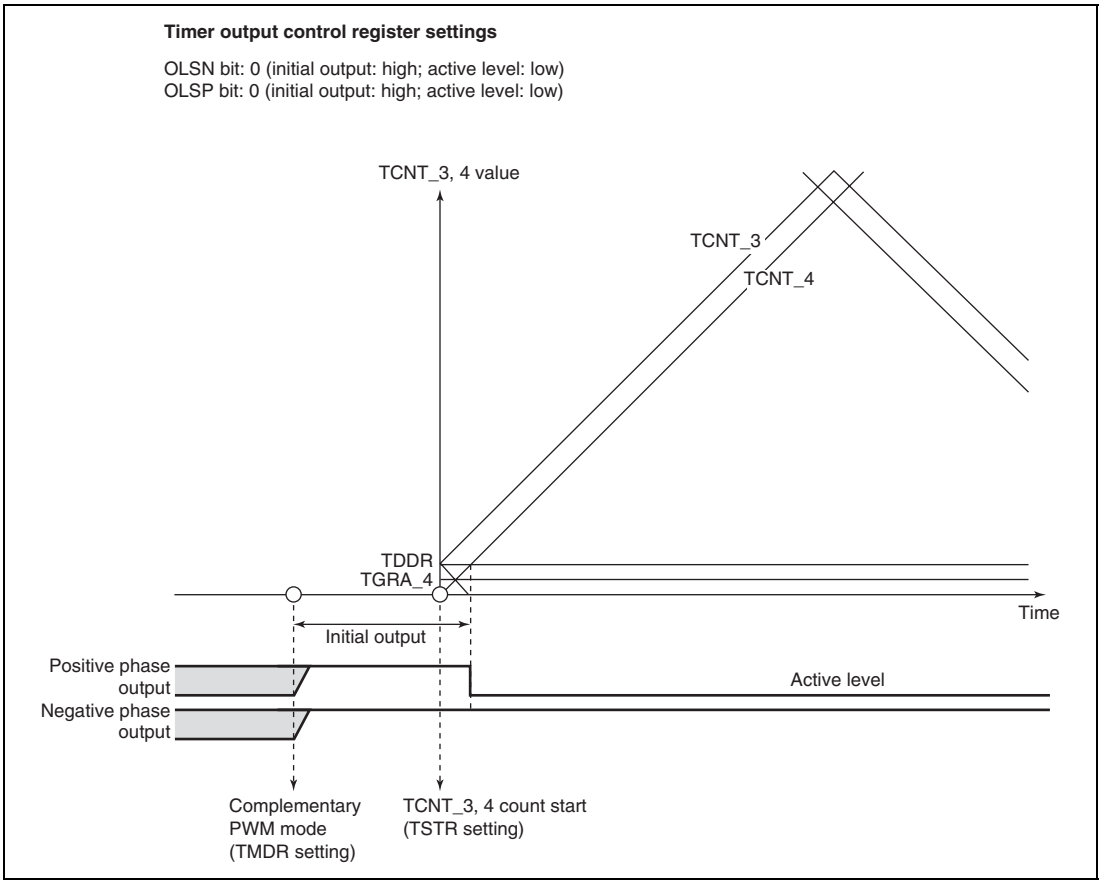


Figure 11.45 Example of Initial Output in Complementary PWM Mode (2)

(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 data register. While TCNTS is counting, data 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 off 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.

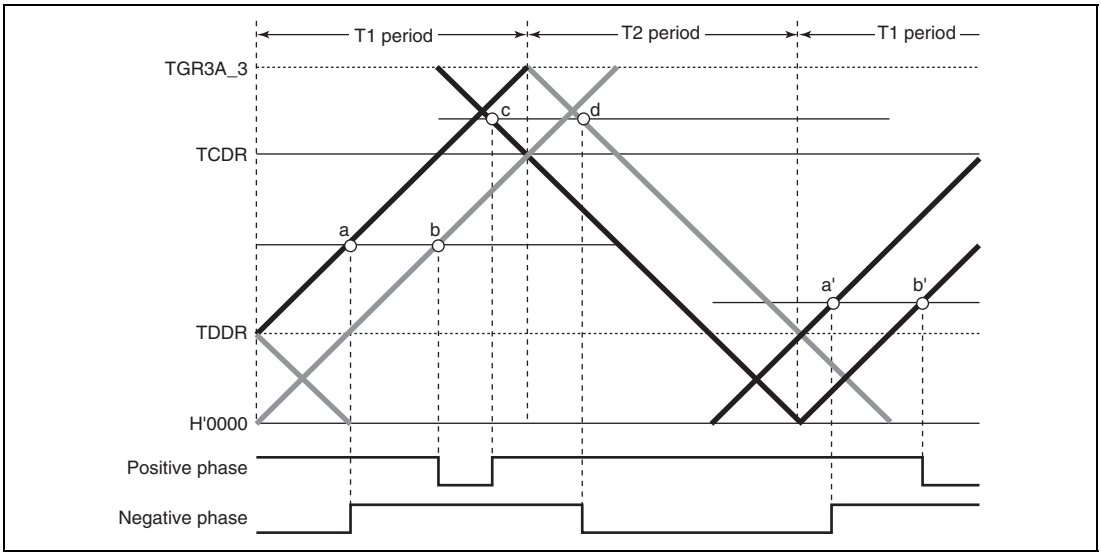


Figure 11.46 Example of Complementary PWM Mode Waveform Output (1)

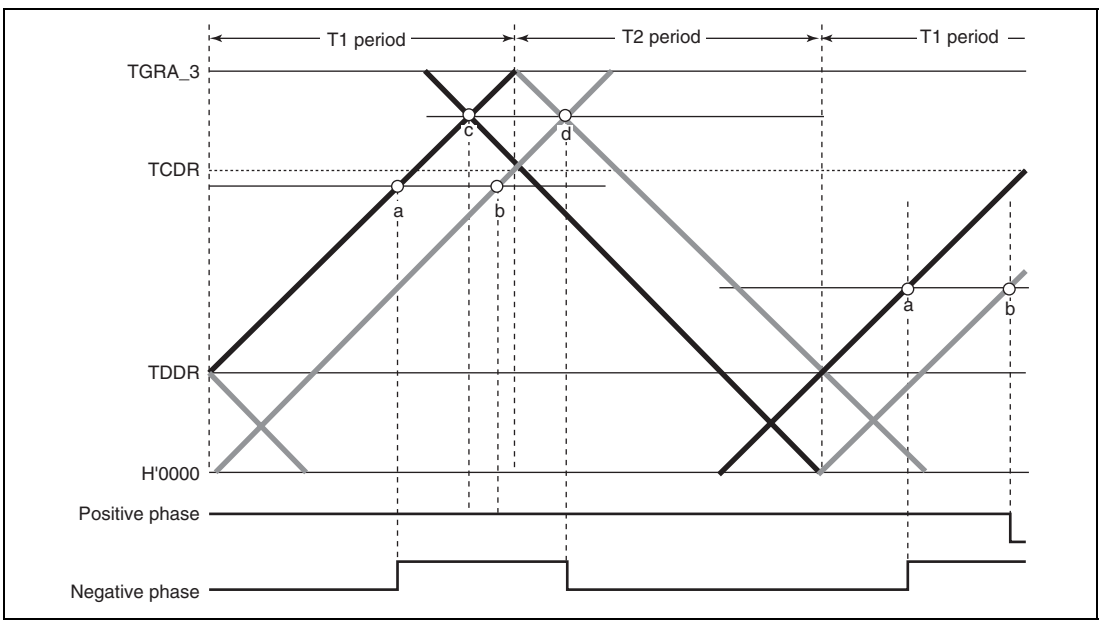


Figure 11.47 Example of Complementary PWM Mode Waveform Output (2)

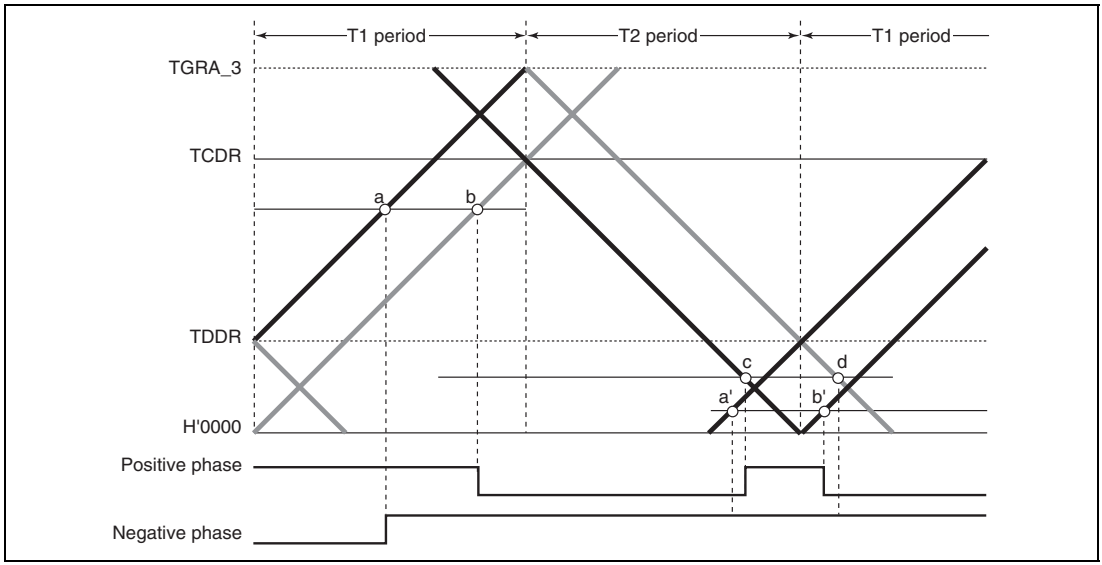


Figure 11.48 Example of Complementary PWM Mode Waveform Output (3)

(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 data 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 data 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.

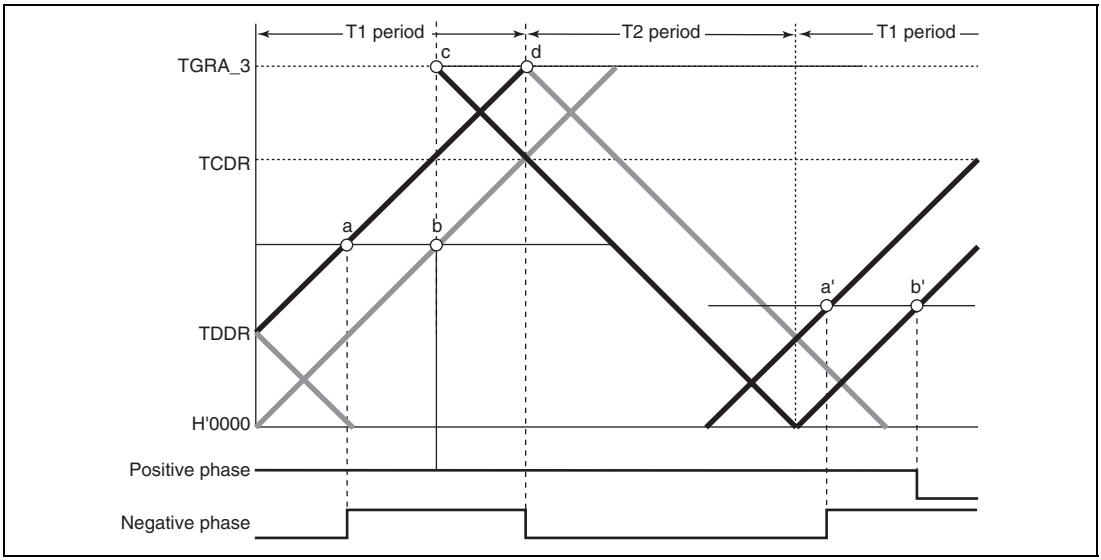


Figure 11.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

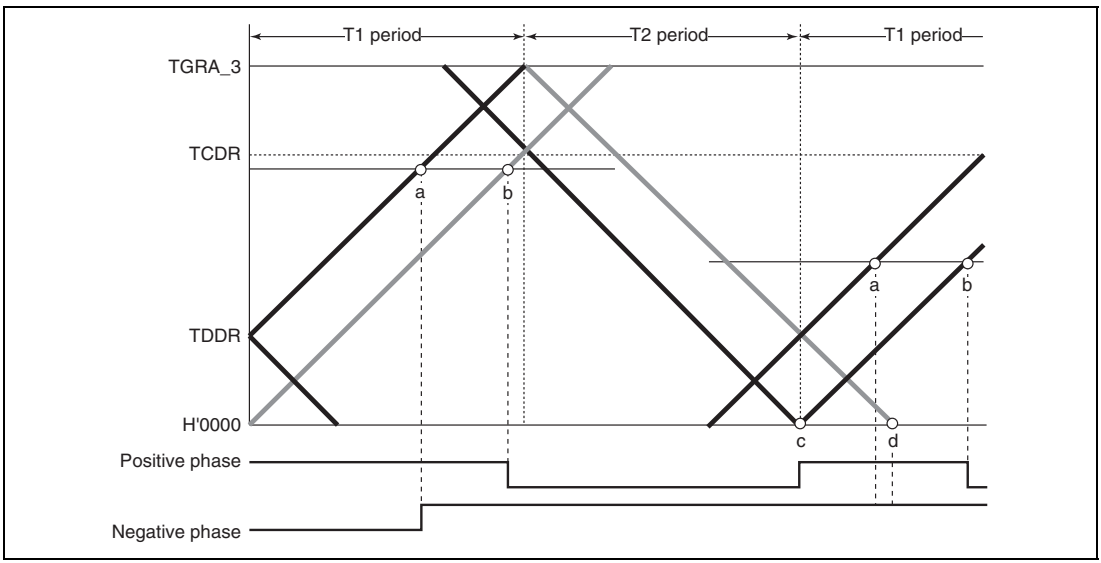


Figure 11.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

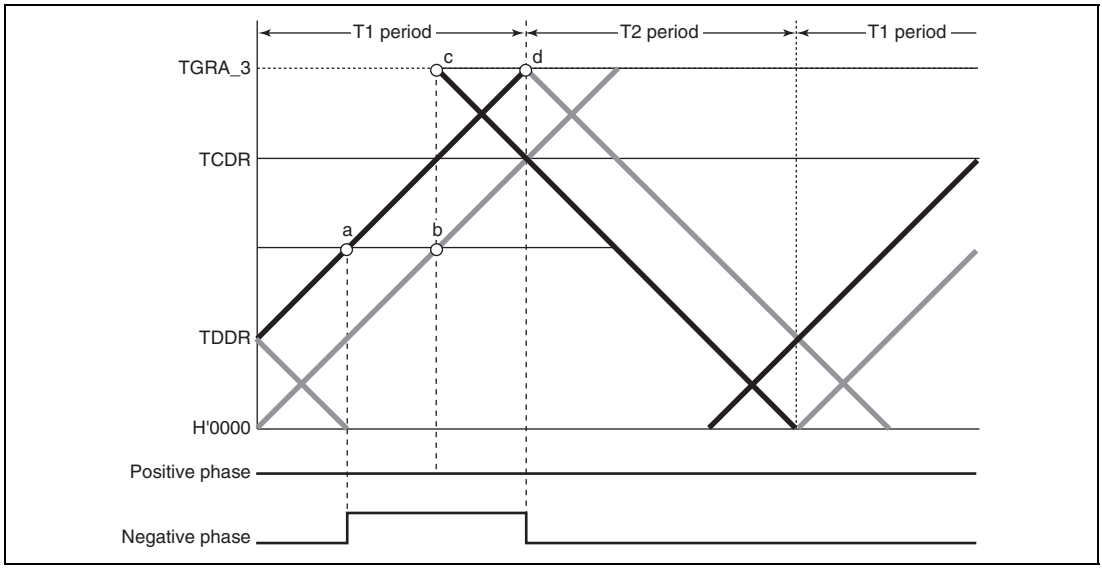


Figure 11.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

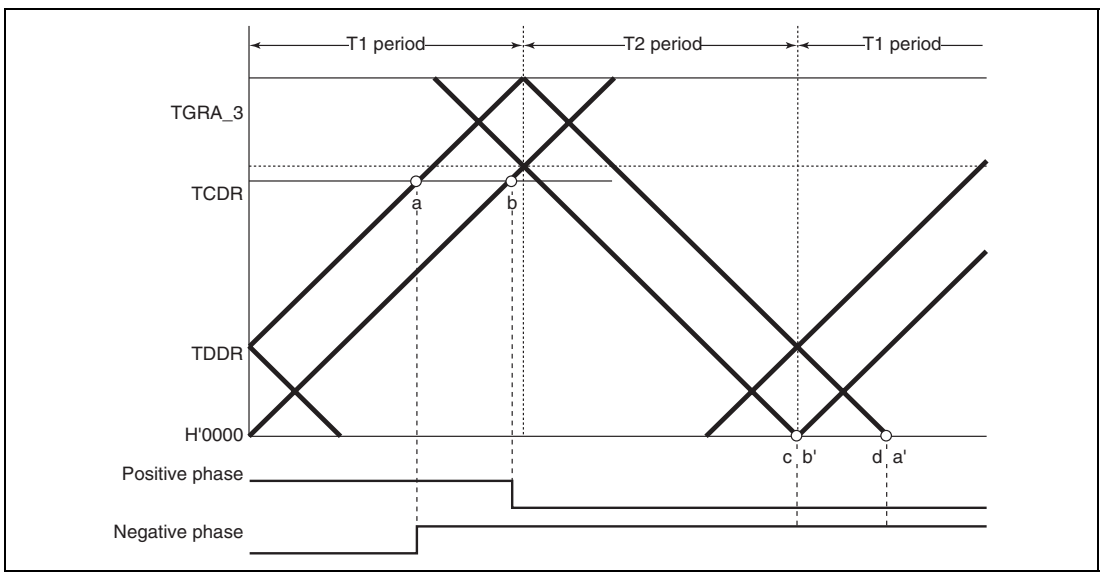


Figure 11.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

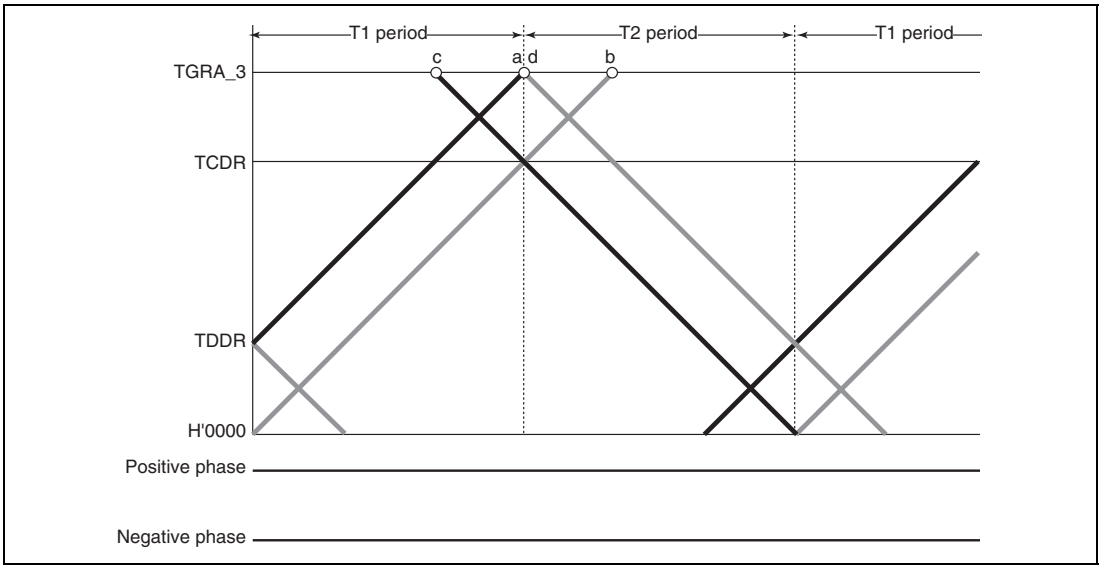


Figure 11.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(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.

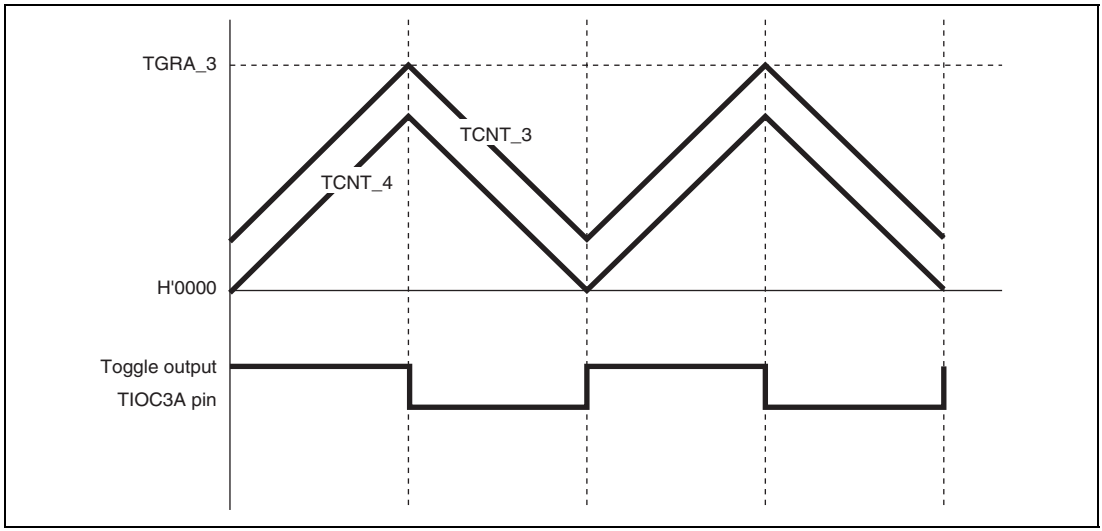


Figure 11.54 Example of Toggle Output Waveform Synchronized with PWM Output

(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.

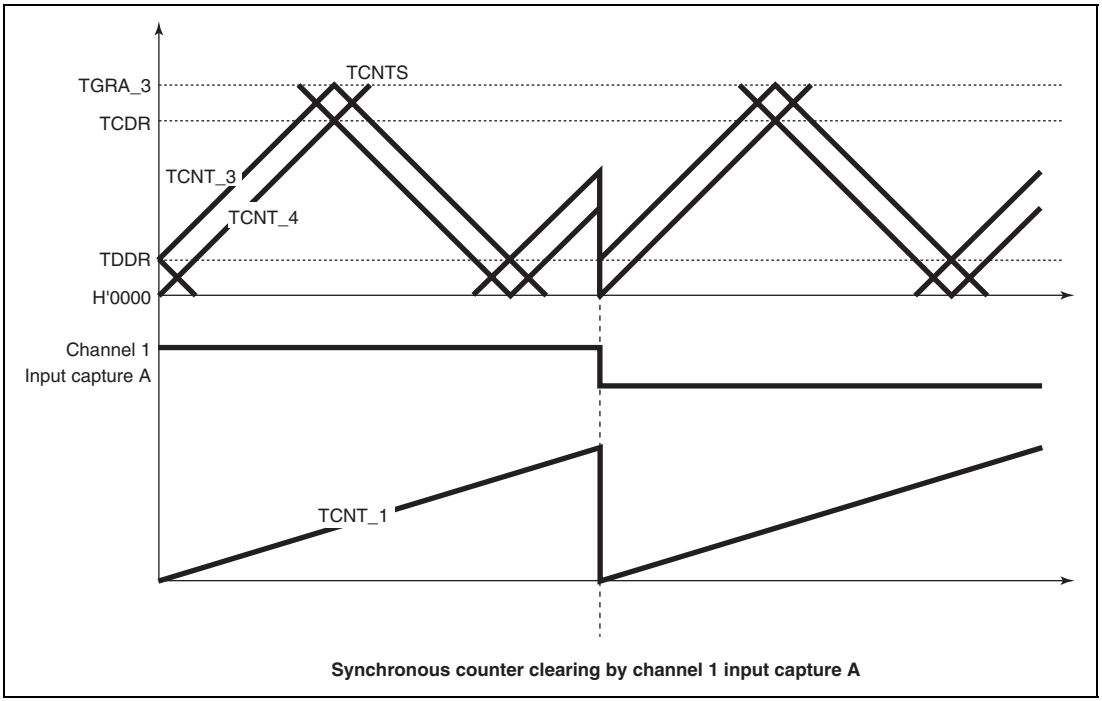


Figure 11.55 Counter Clearing Synchronized with Another Channel

(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.

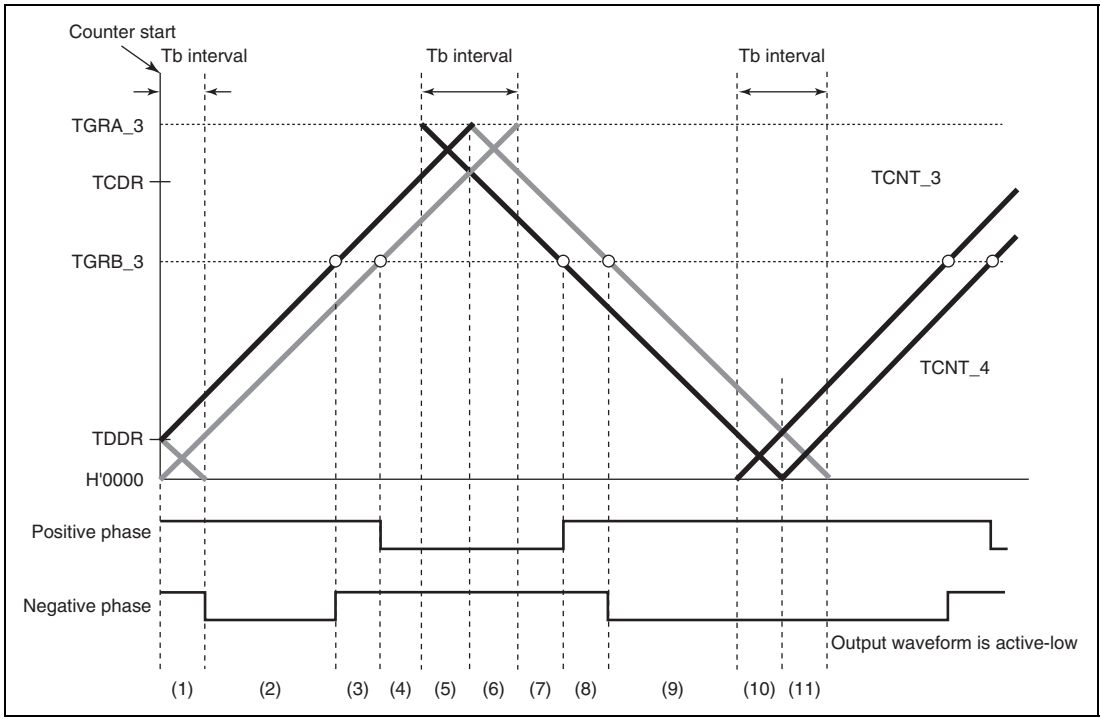


Figure 11.56 Timing for Synchronous Counter Clearing

- 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.

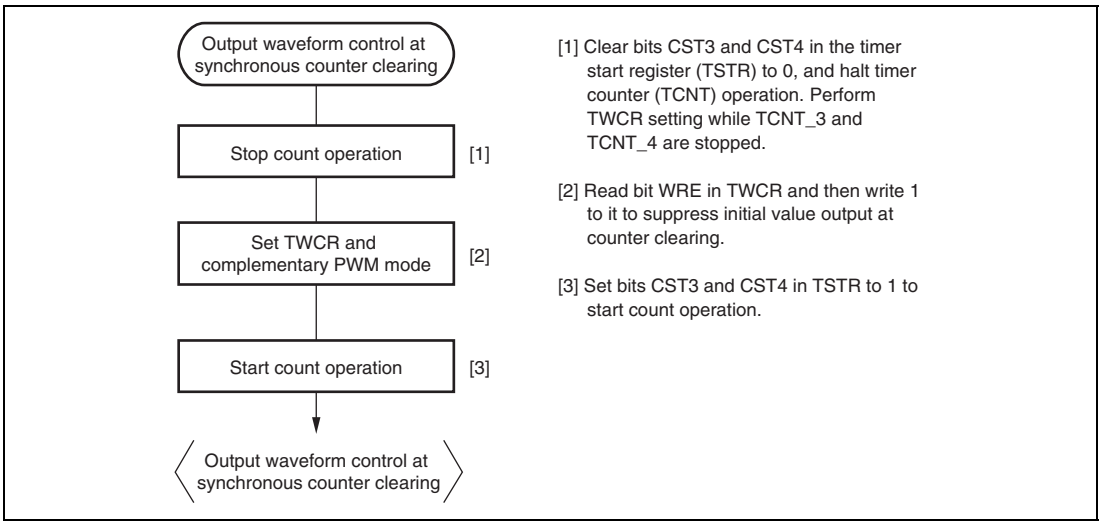


Figure 11.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- 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.

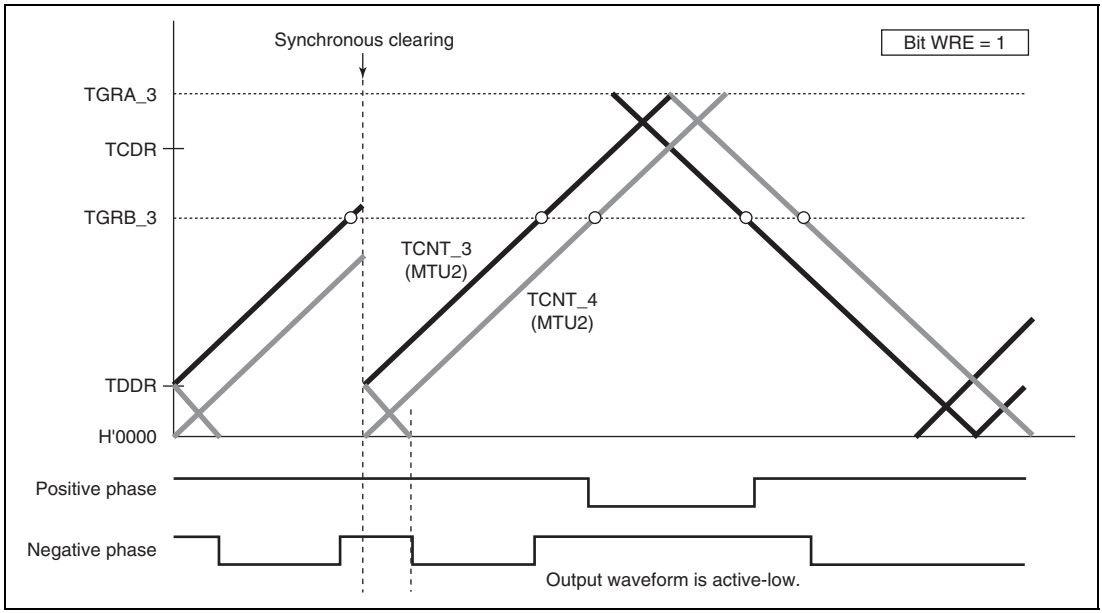


Figure 11.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 11.56; Bit WRE of TWCR in MTU2 is 1)

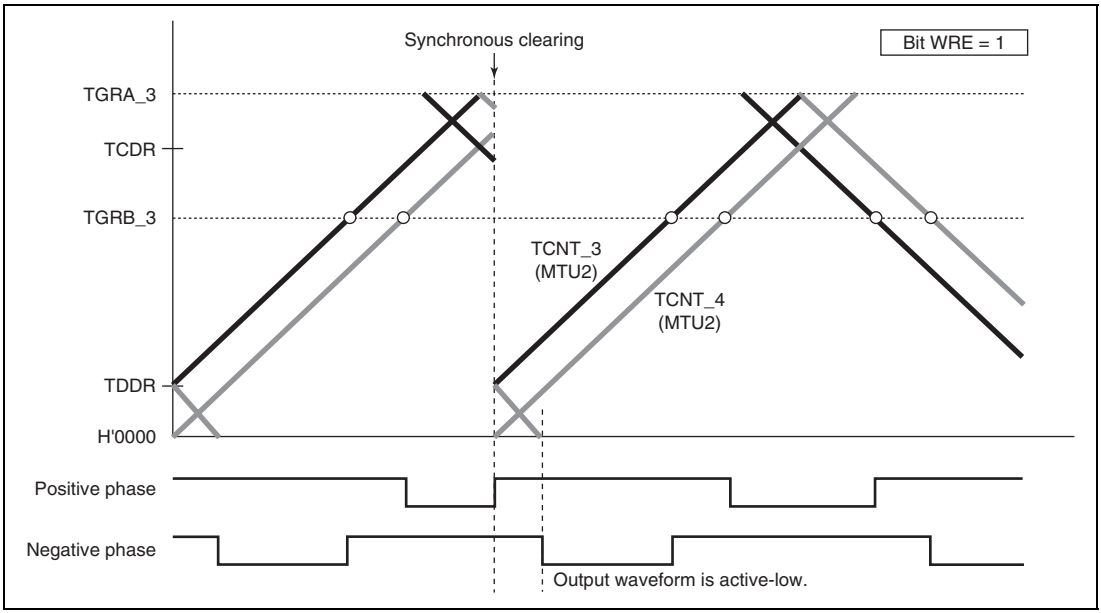


Figure 11.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 11.56; Bit WRE of TWCR in MTU2 is 1)

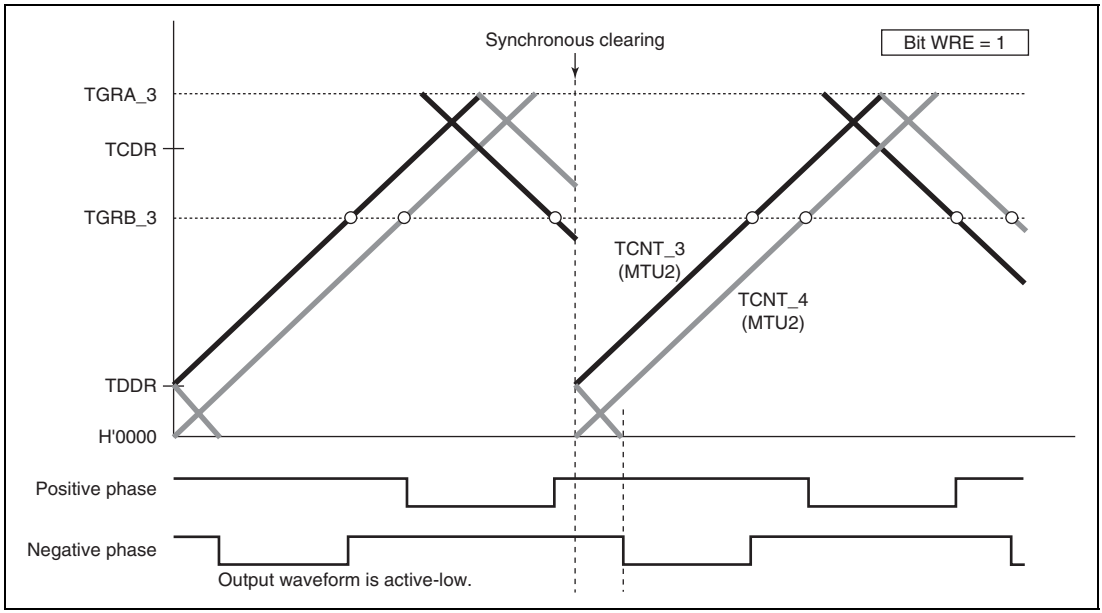


Figure 11.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 11.56; Bit WRE of TWCR is 1)

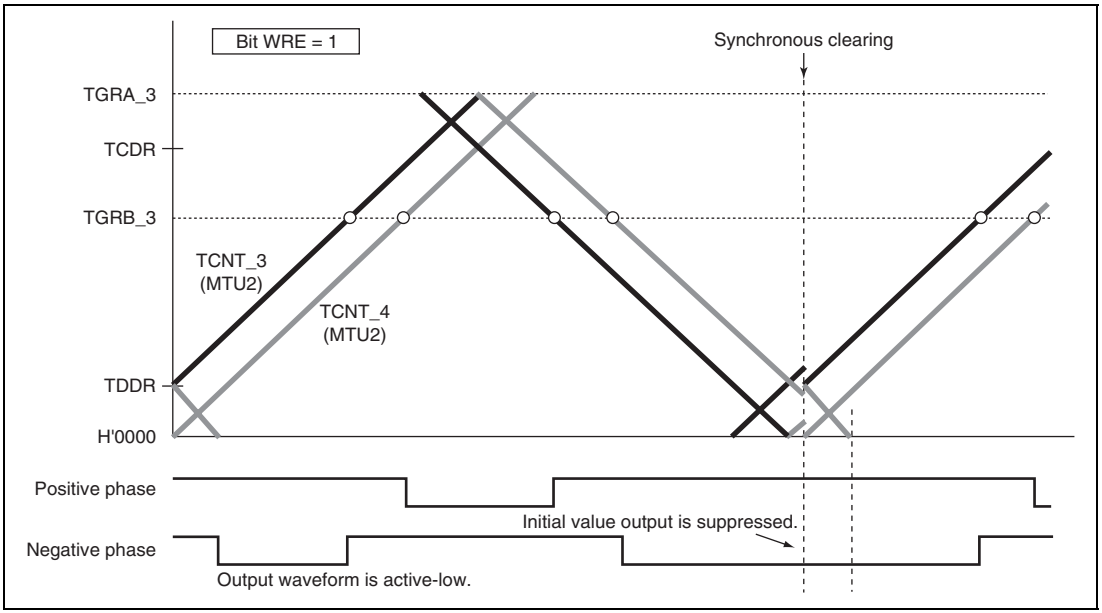


Figure 11.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 11.56; Bit WRE of TWCR is 1)

(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.

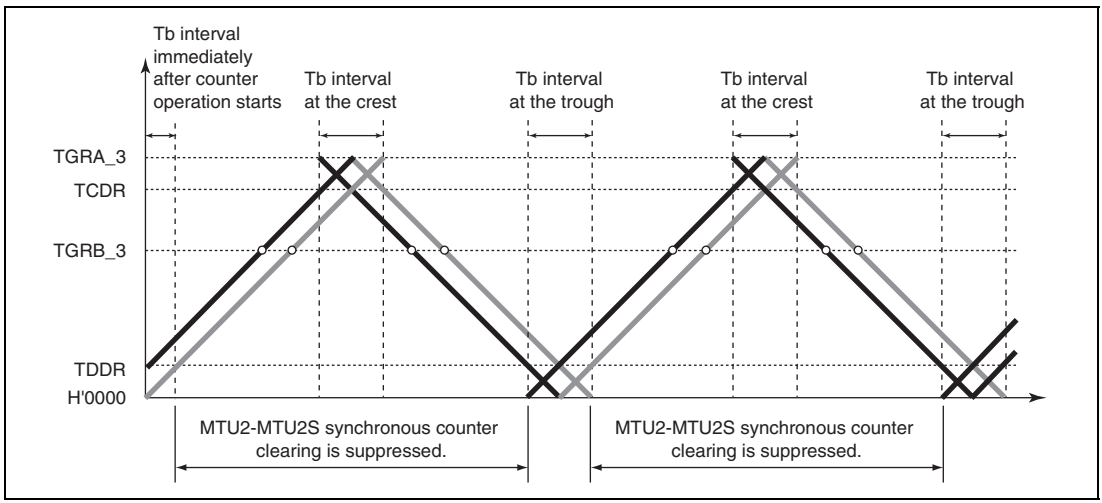


Figure 11.62 MTU2-MTU2S Synchronous Clearing-Suppressed Interval Specified by SCC Bit in TWCR

- 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.

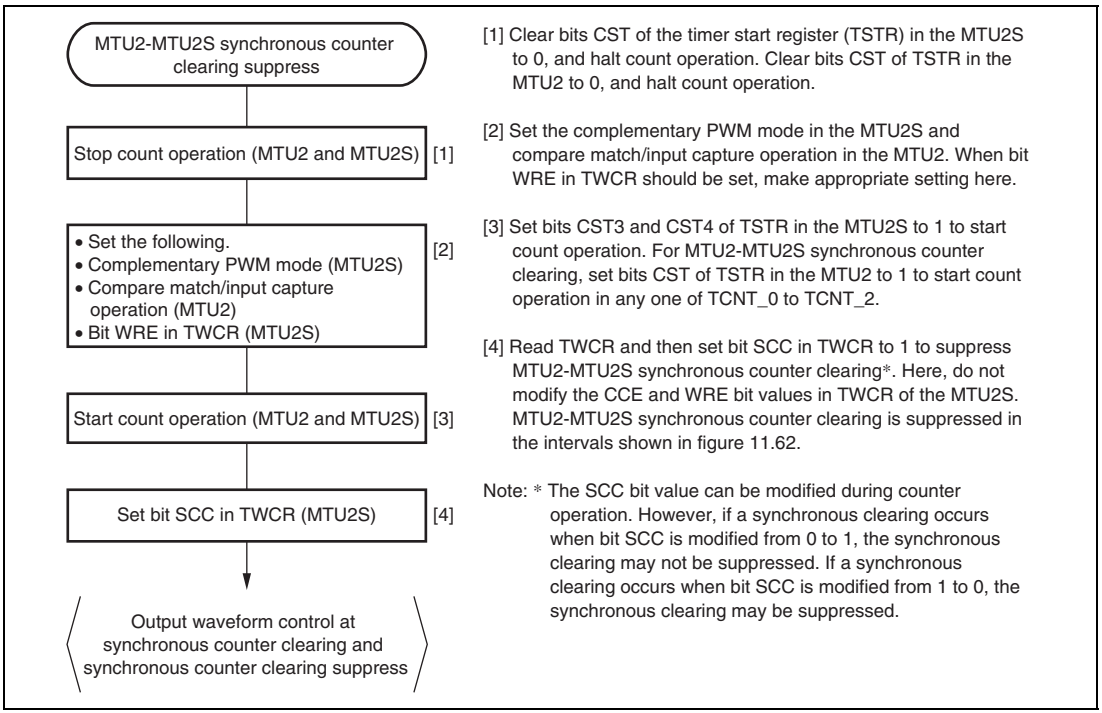


Figure 11.63 Example of Procedure for Suppressing MTU2-MTU2S Synchronous Counter Clearing

- 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.

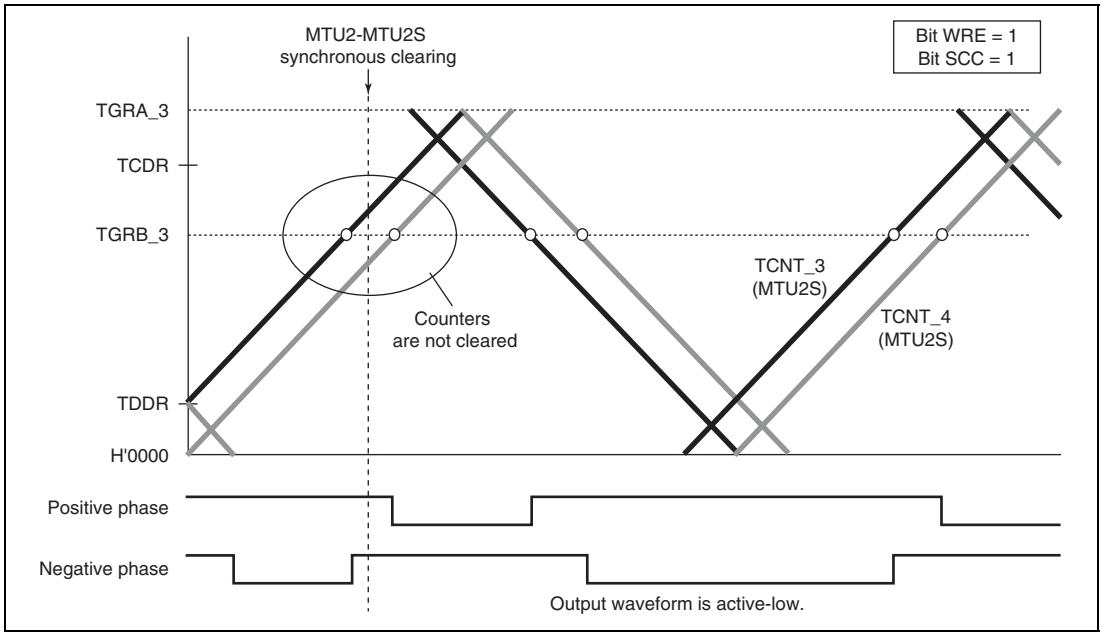


Figure 11.64 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

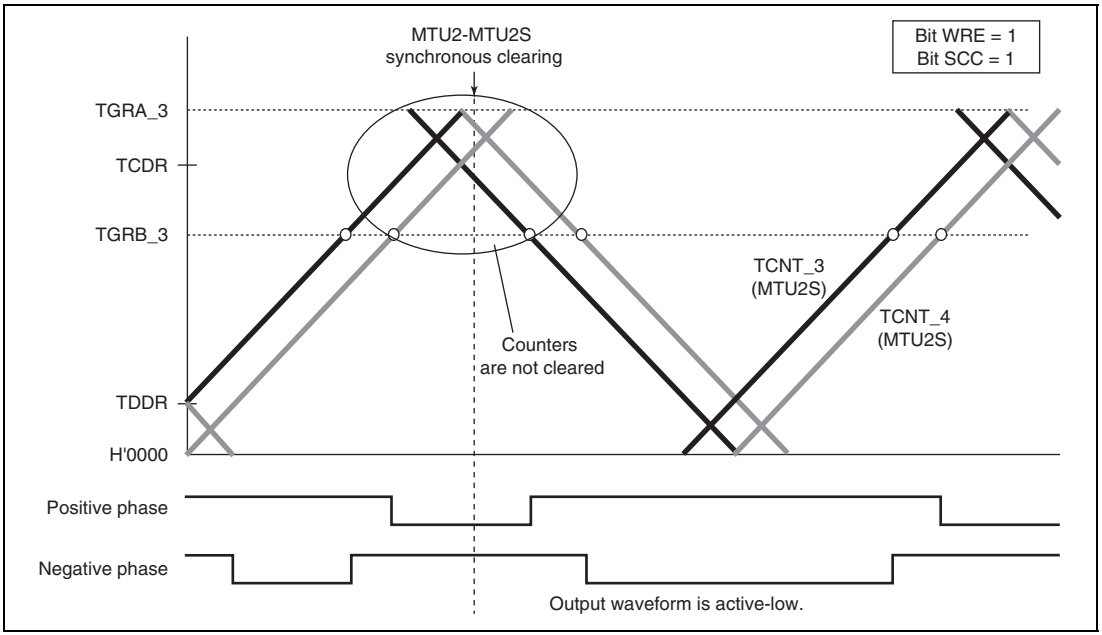


Figure 11.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

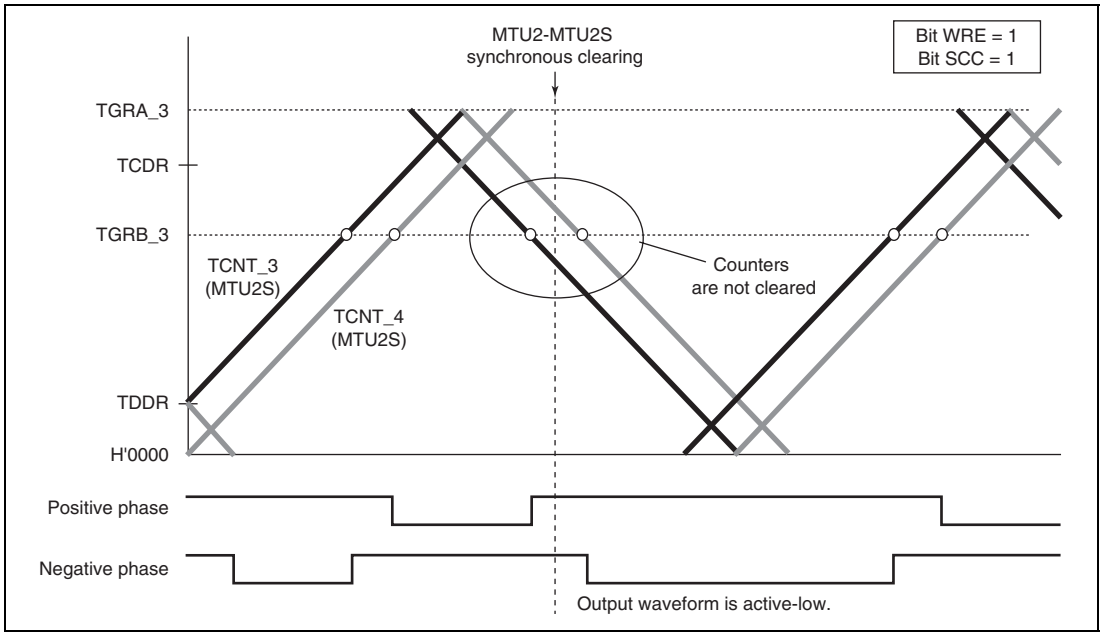


Figure 11.66 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

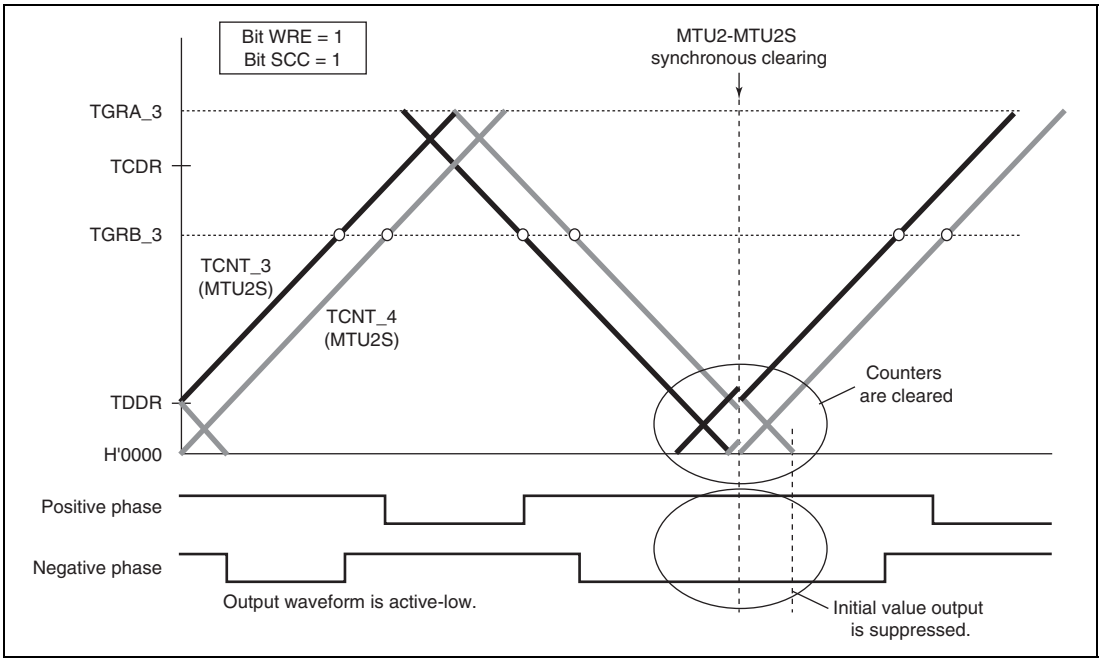


Figure 11.67 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 11.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

(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.

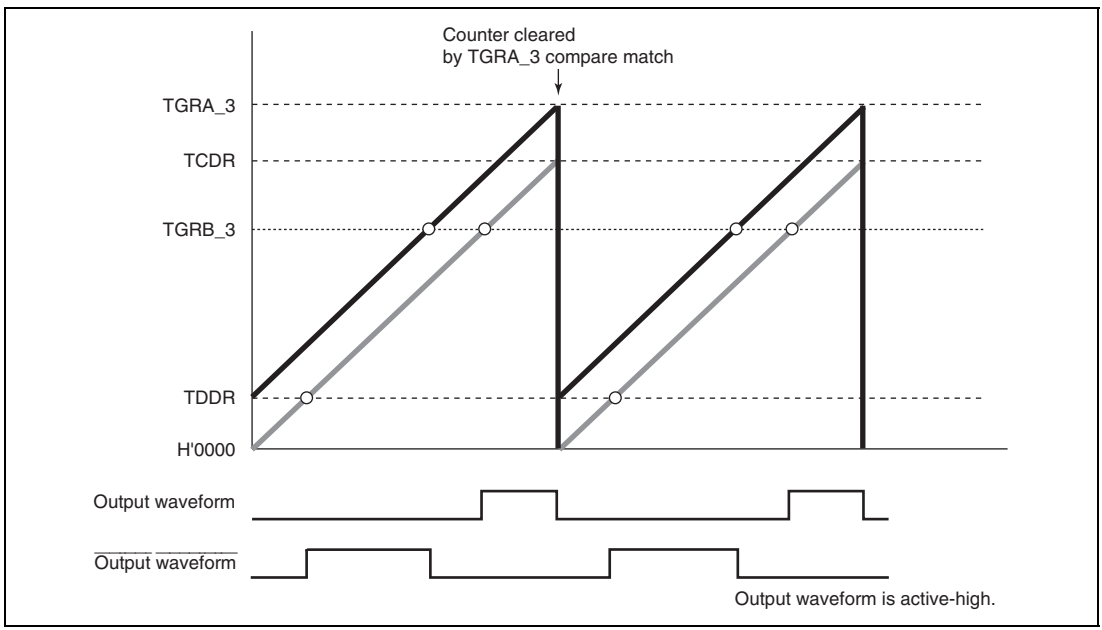


Figure 11.68 Example of Counter Clearing Operation by TGRA_3 Compare Match

(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.

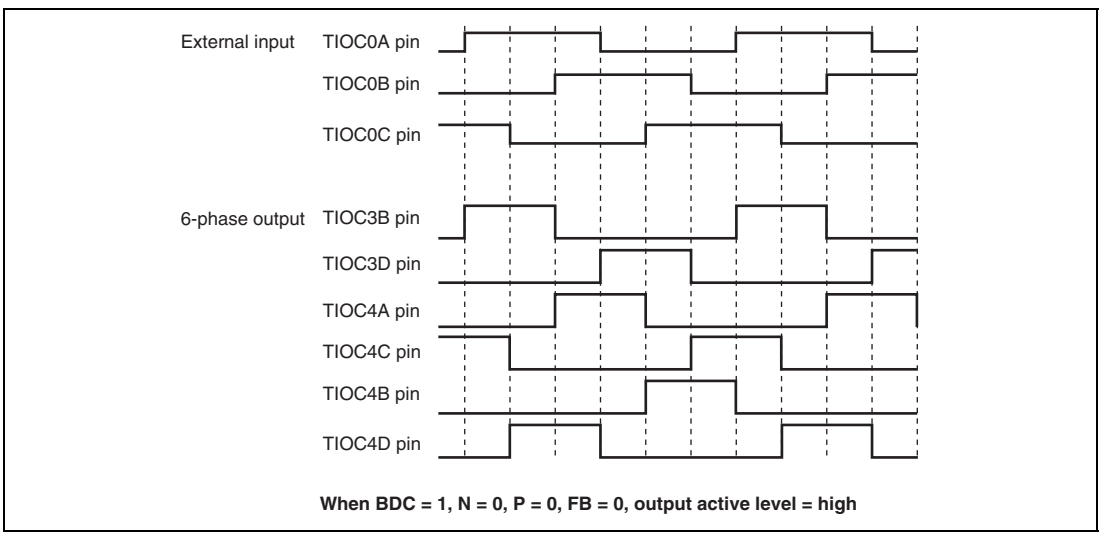


Figure 11.69 Example of Output Phase Switching by External Input (1)

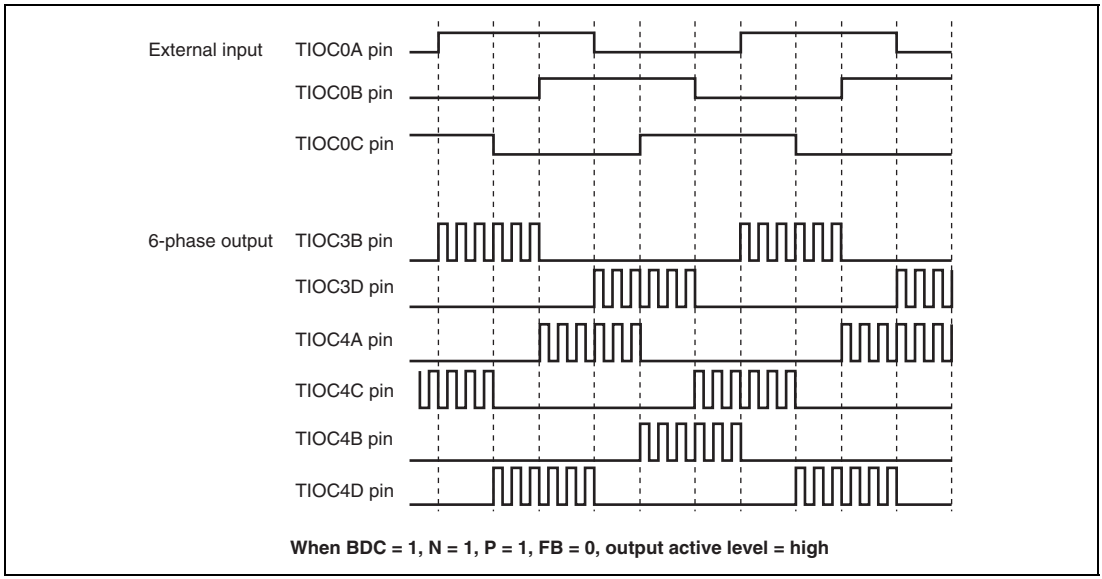


Figure 11.70 Example of Output Phase Switching by External Input (2)

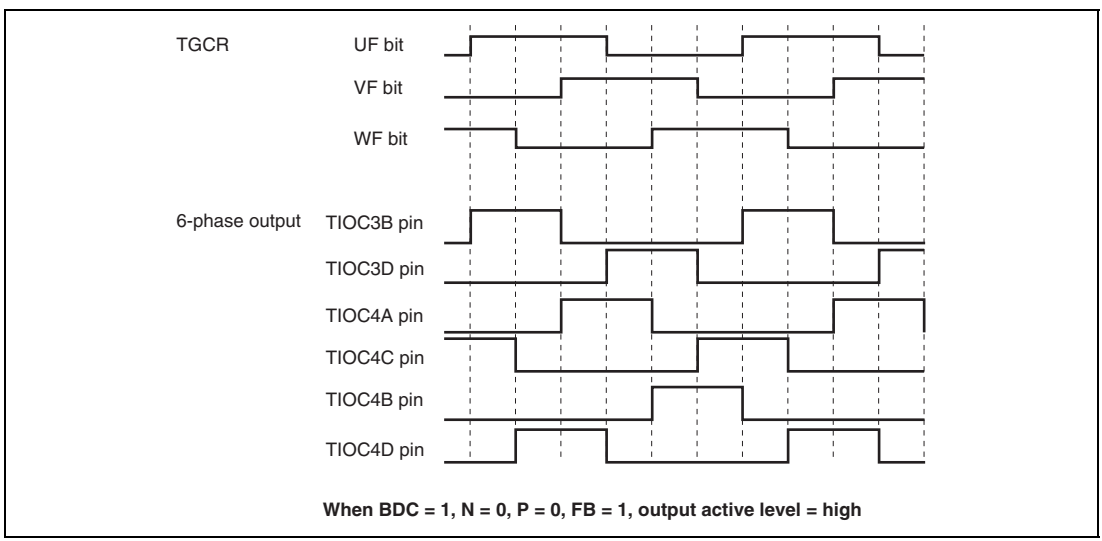


Figure 11.71 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

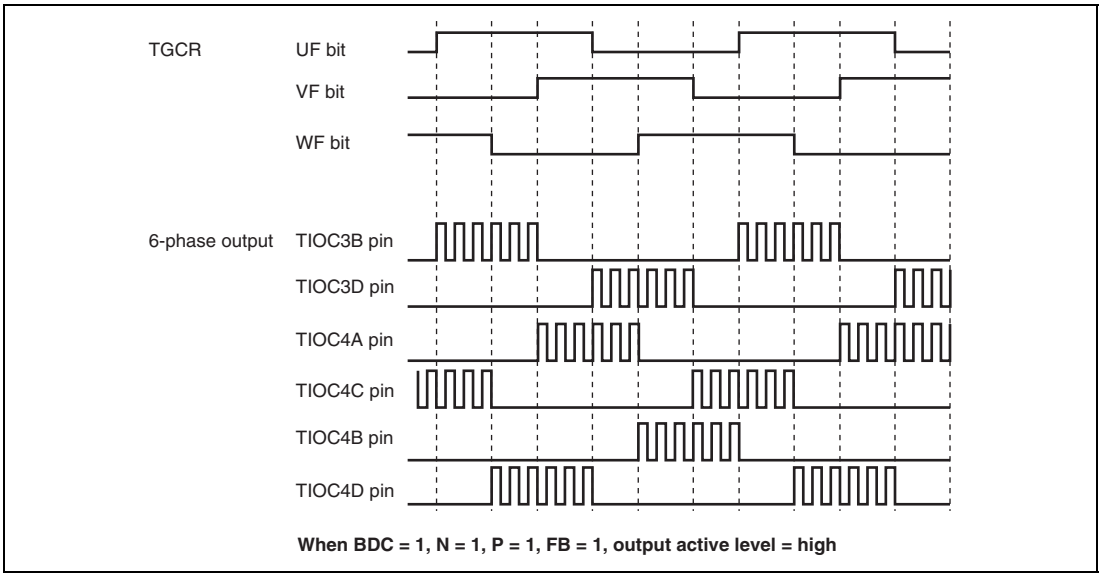


Figure 11.72 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(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.

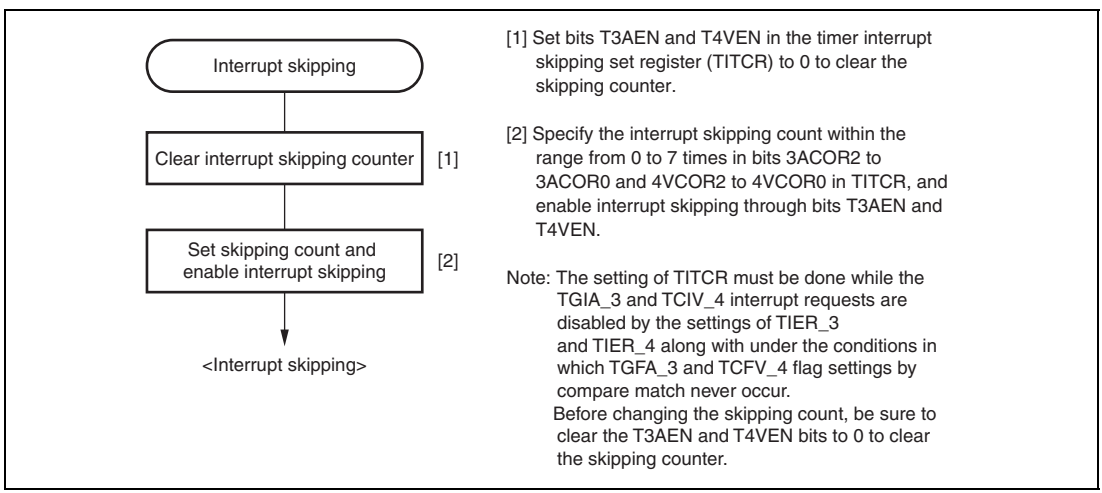


Figure 11.73 Example of Interrupt Skipping Operation Setting Procedure

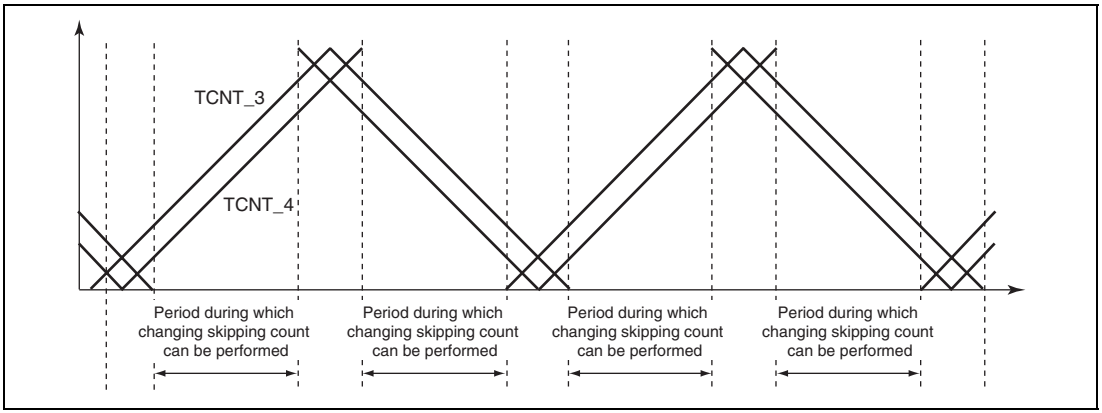


Figure 11.74 Periods during which Interrupt Skipping Count can be Changed

(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).

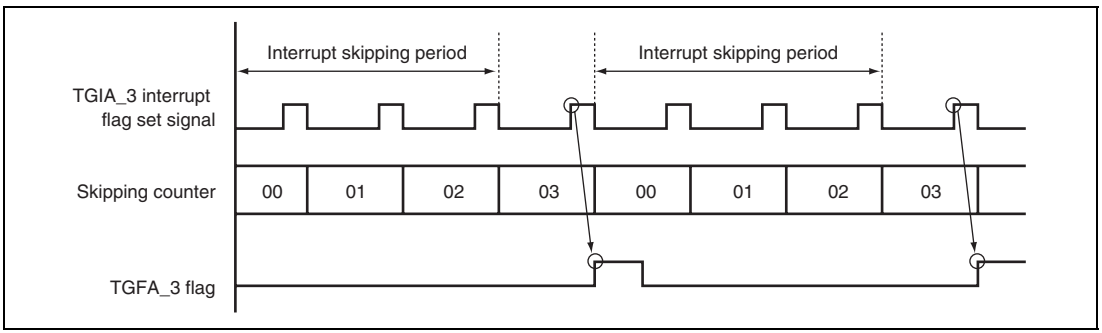


Figure 11.75 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

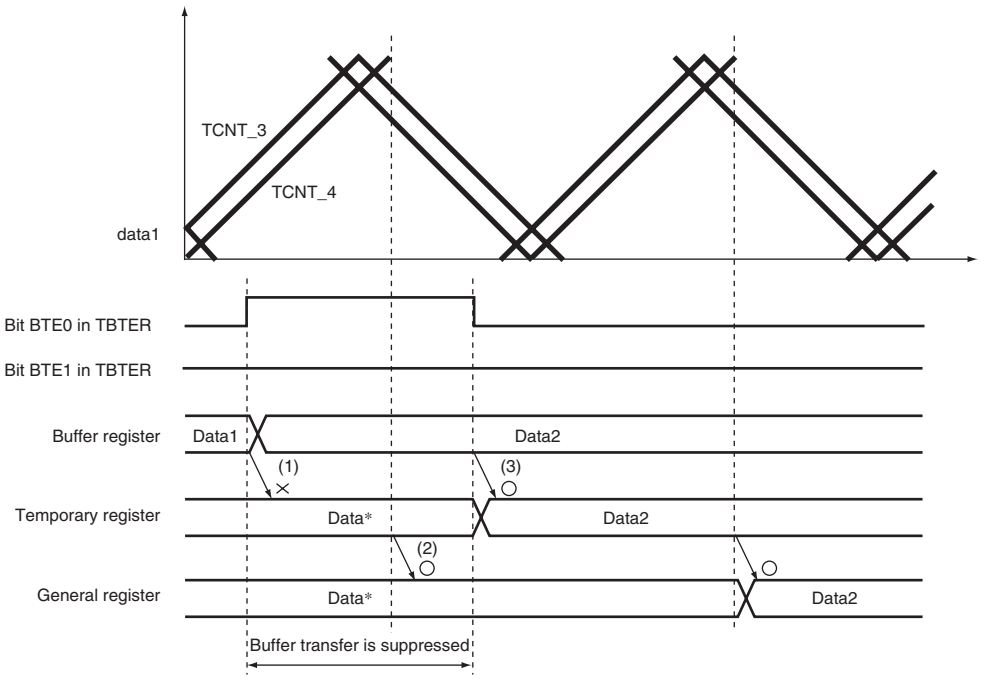
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

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.

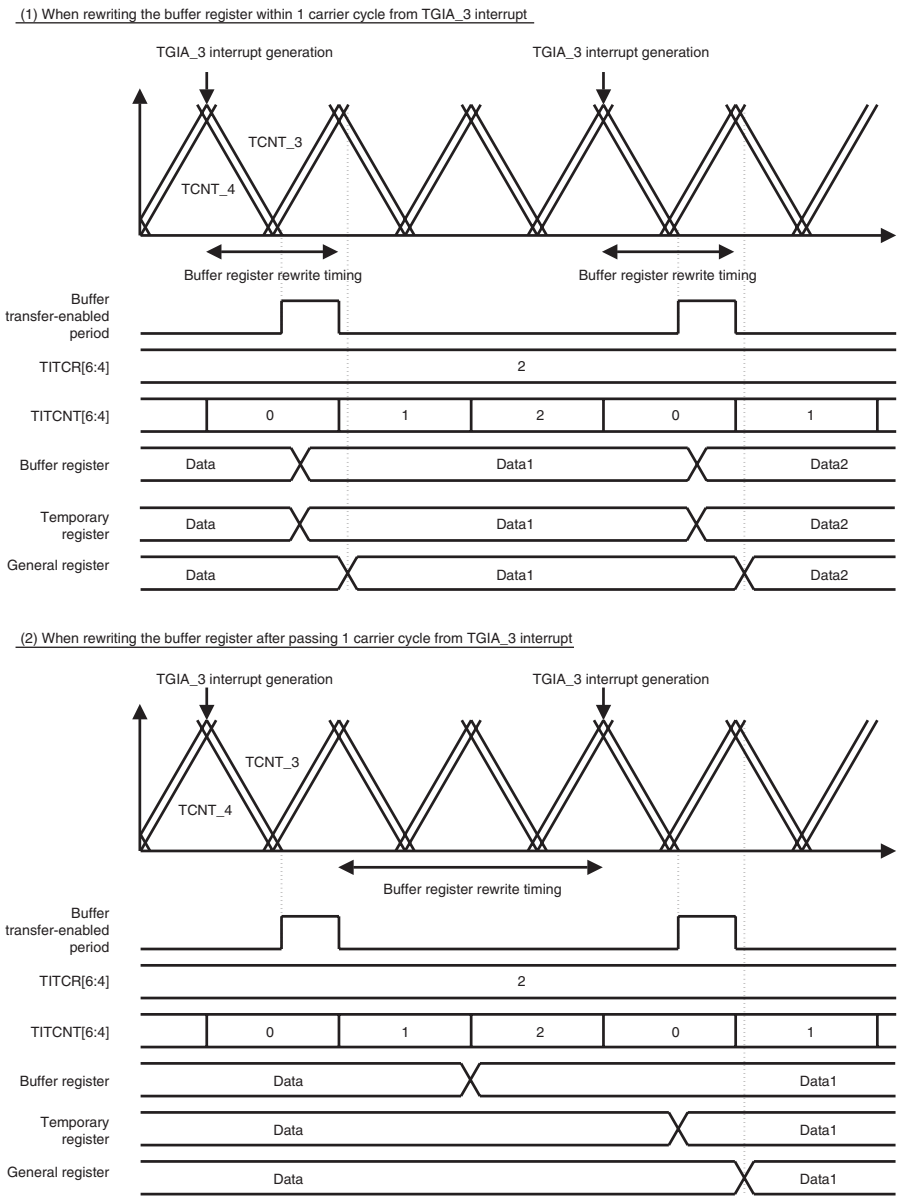


[Legend]

(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)

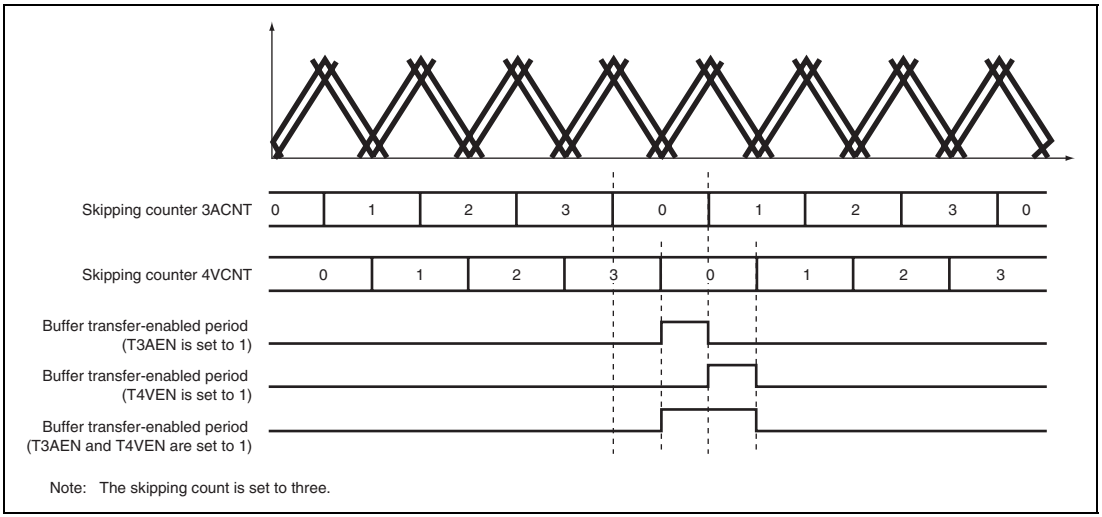


Figure 11.78 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(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.

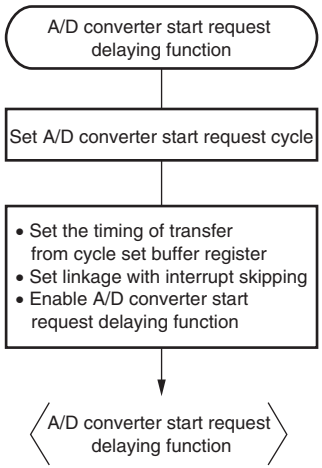
11.4.9 A/D Converter Start Request Delaying Function

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.

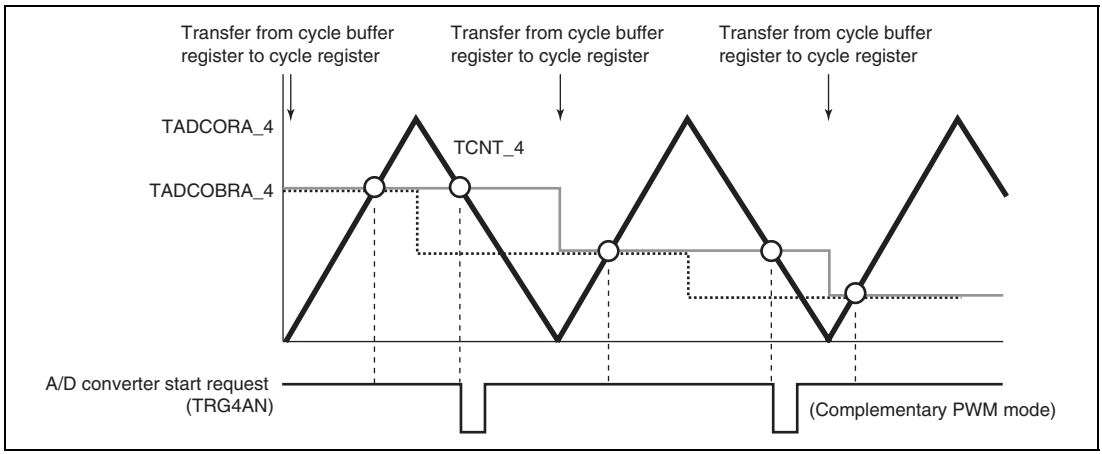


Figure 11.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- 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.

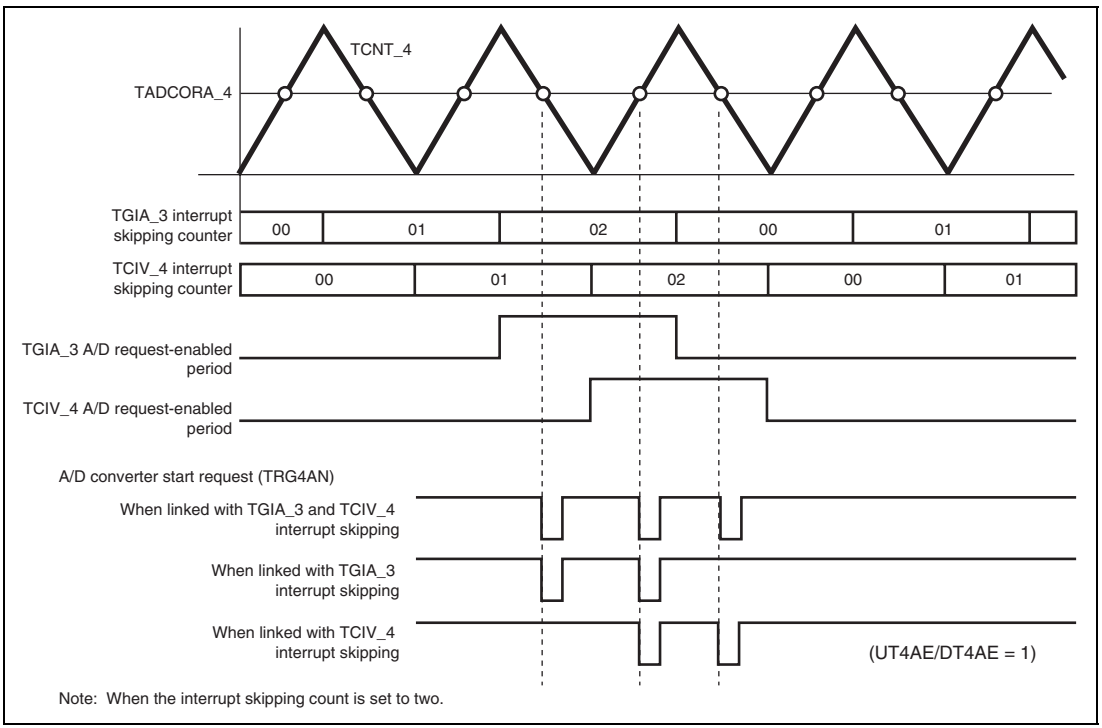


Figure 11.81 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

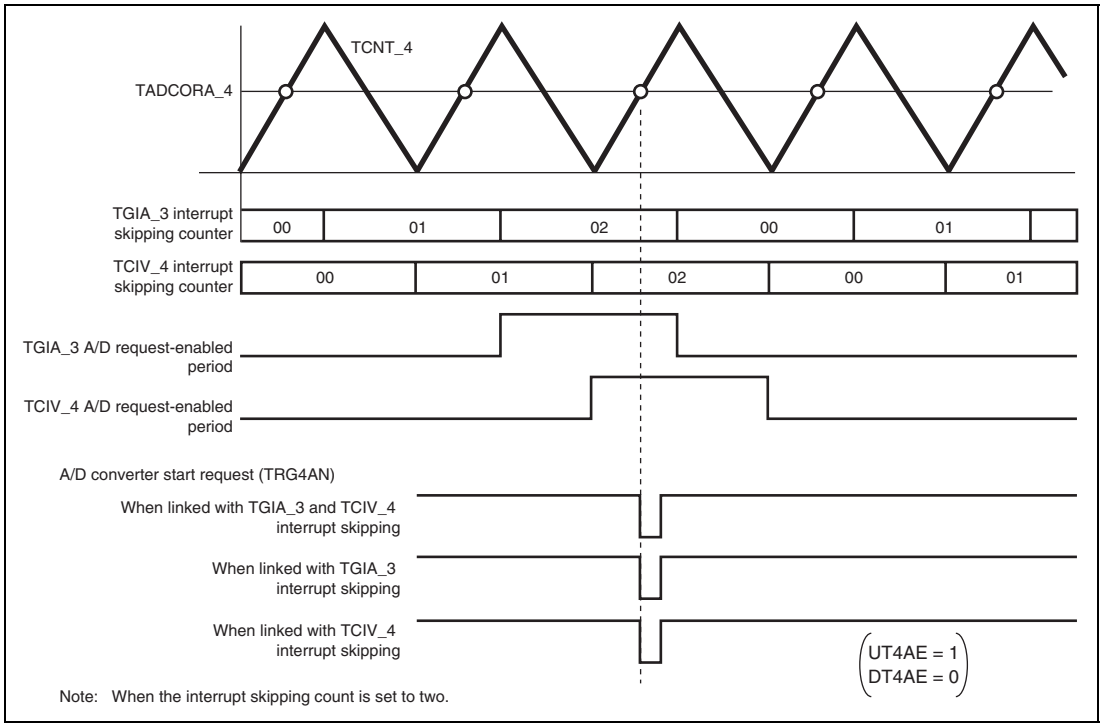


Figure 11.82 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

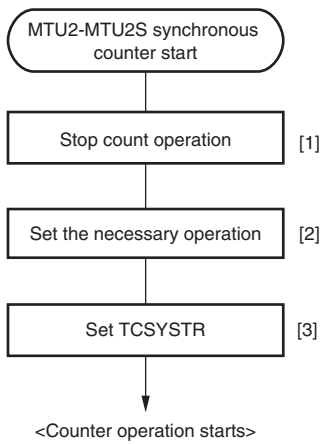
11.4.10 MTU2-MTU2S Synchronous Operation

(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

Figure 11.84 shows an example of synchronous counter start operation.

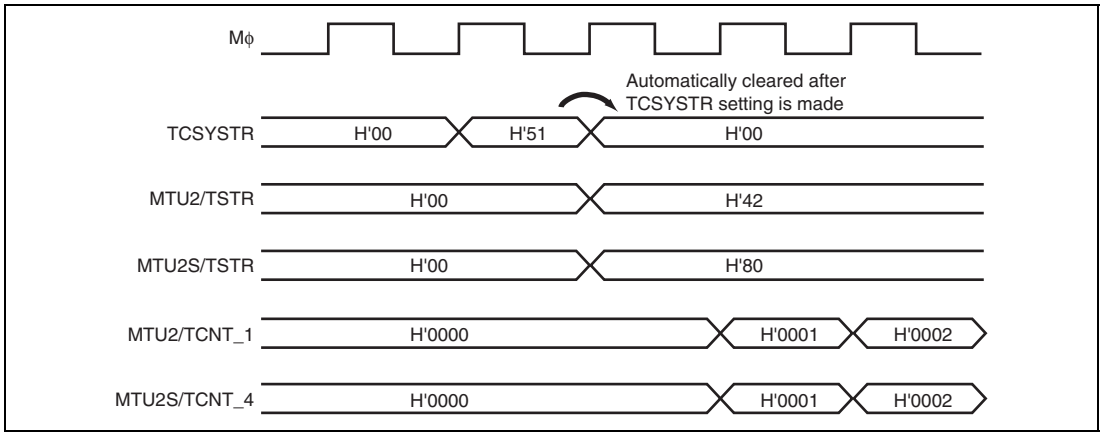


Figure 11.84 Example of Synchronous Counter Start Operation

(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.

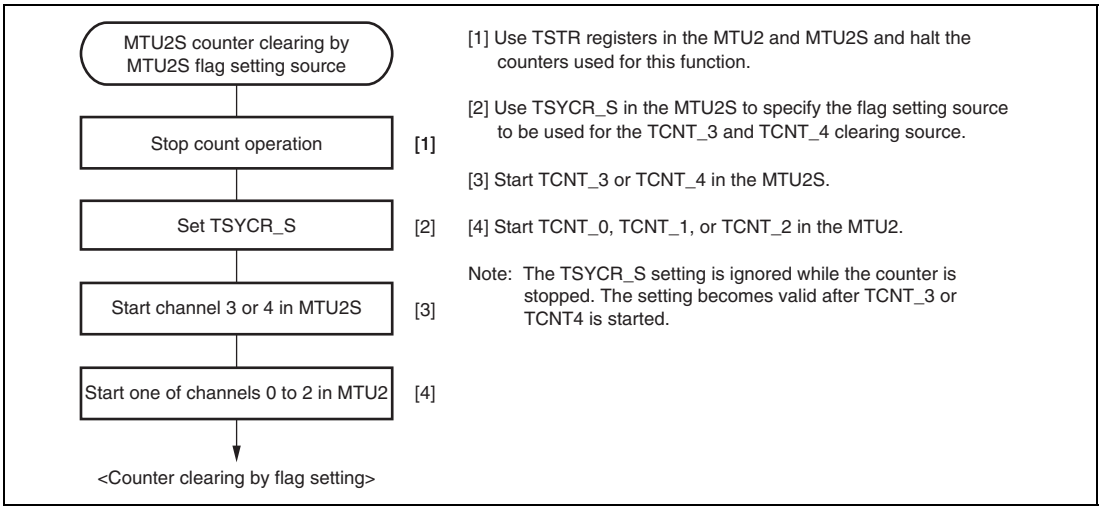


Figure 11.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

(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.

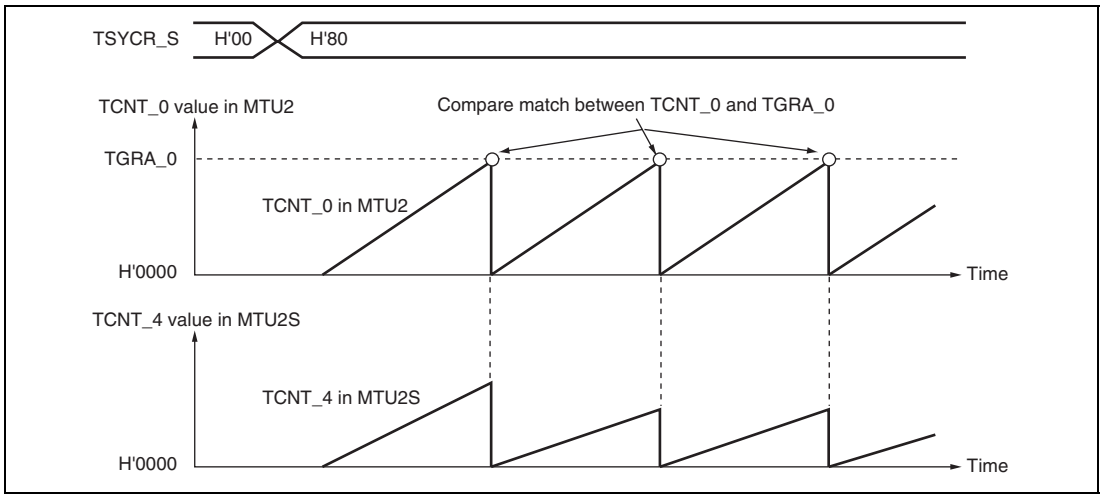


Figure 11.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)

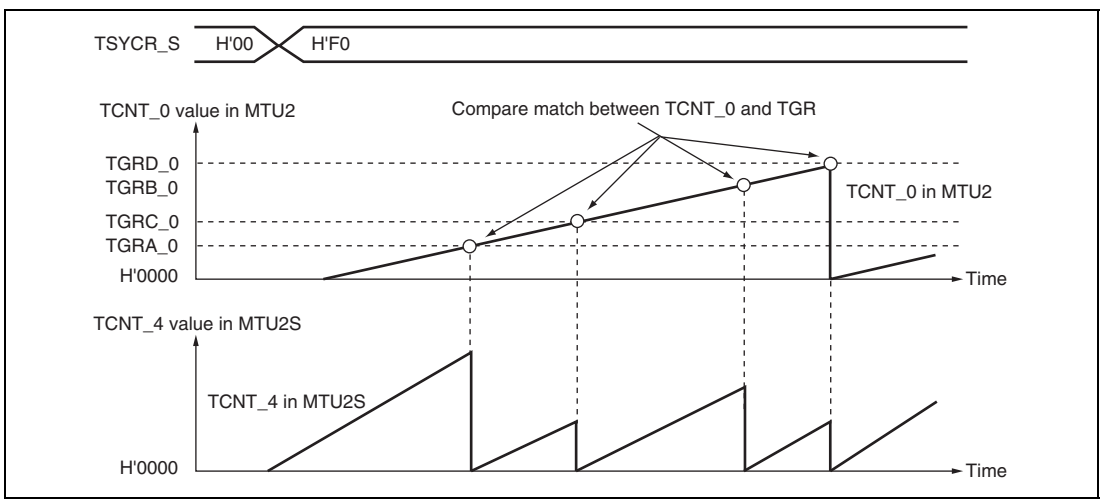


Figure 11.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)

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

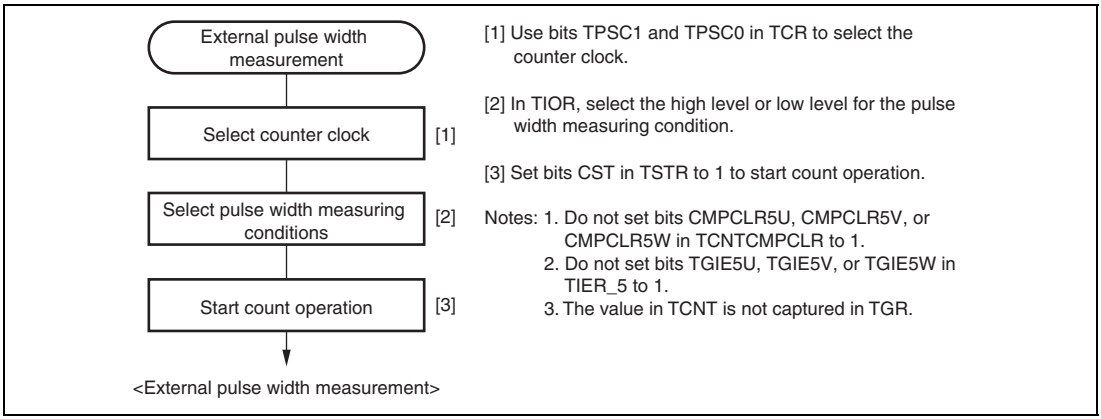


Figure 11.87 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

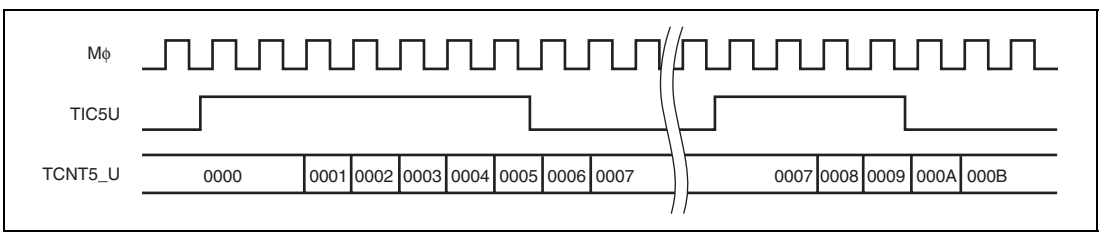


Figure 11.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)

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.

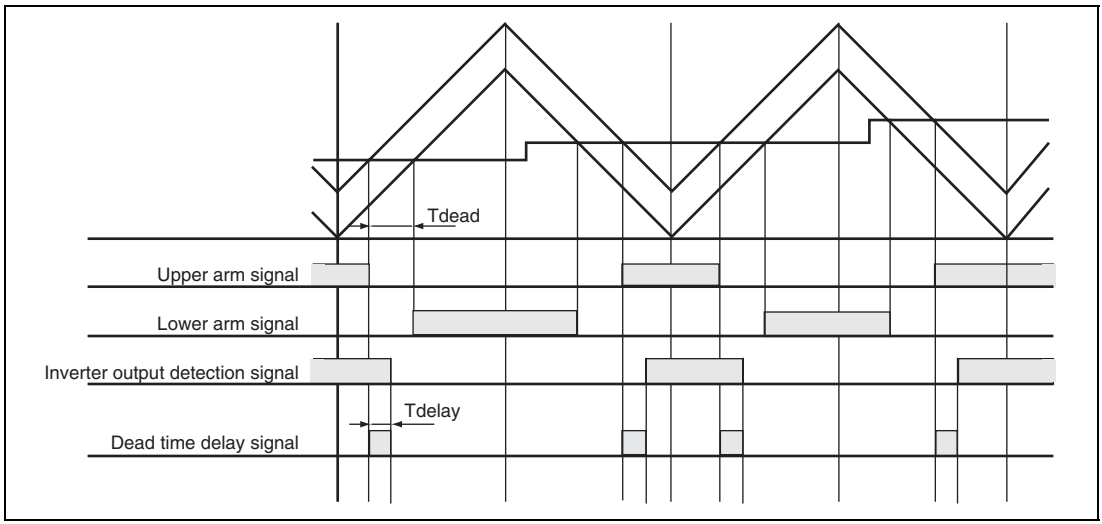
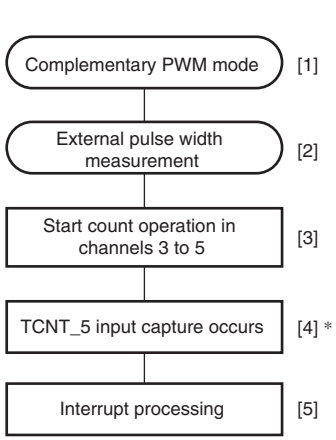


Figure 11.89 Delay in Dead Time in Complementary PWM Operation

(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

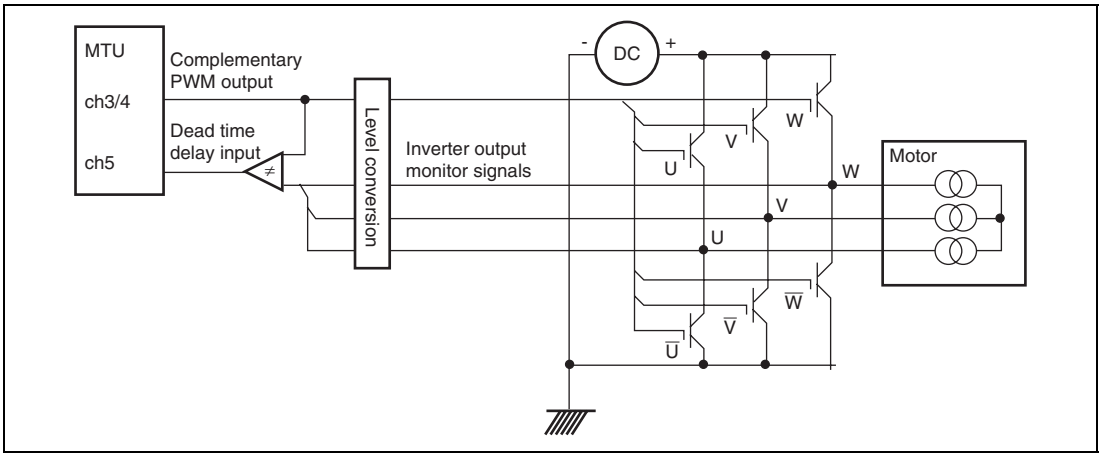


Figure 11.91 Example of Motor Control Circuit Configuration

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).

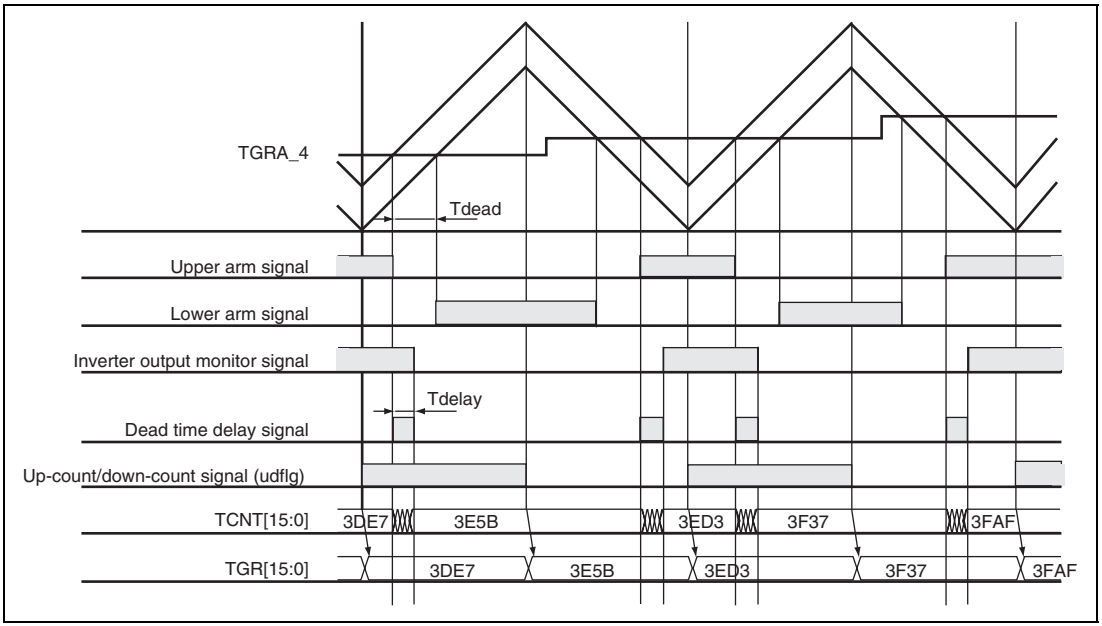


Figure 11.92 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

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 9.4.4, 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).

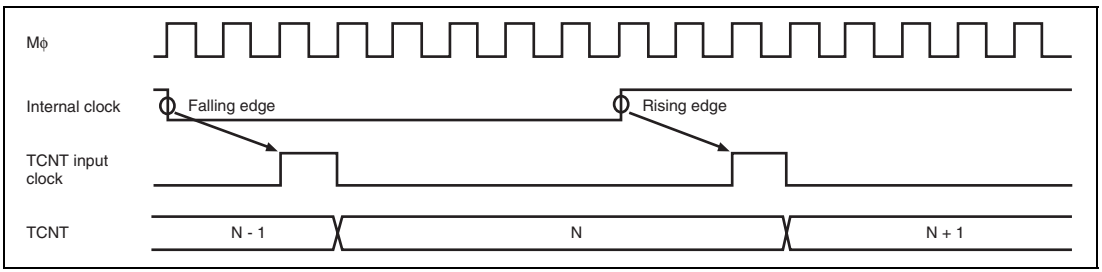


Figure 11.93 Count Timing in Internal Clock Operation (Channels 0 to 4)

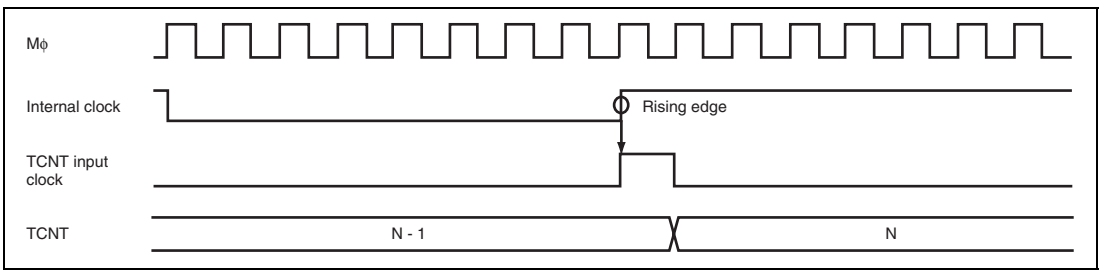


Figure 11.94 Count Timing in Internal Clock Operation (Channel 5)

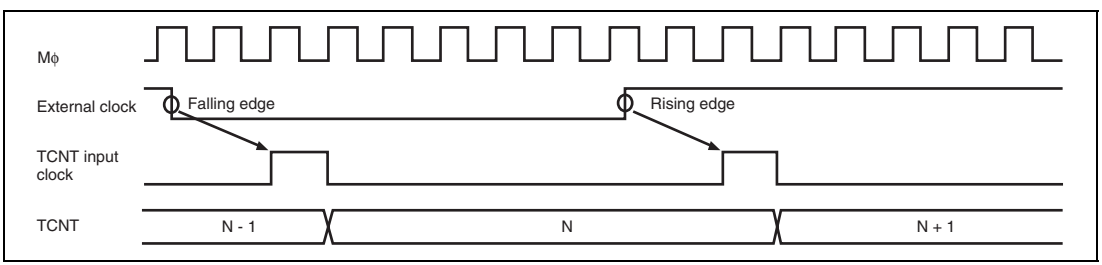


Figure 11.95 Count Timing in External Clock Operation (Channels 0 to 4)

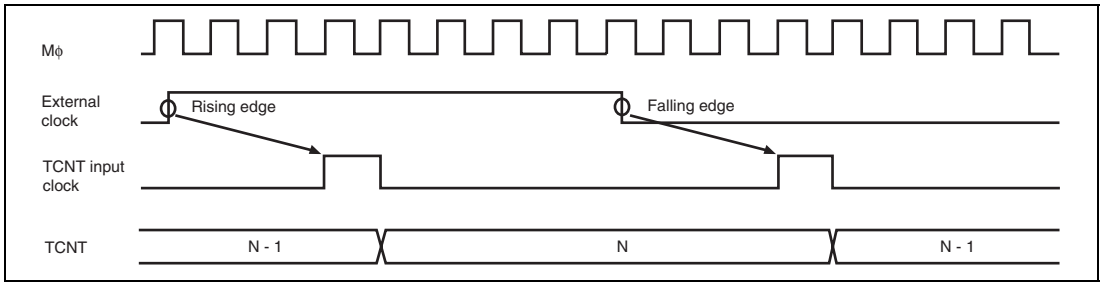


Figure 11.96 Count Timing in External Clock Operation (Phase Counting Mode)

(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).

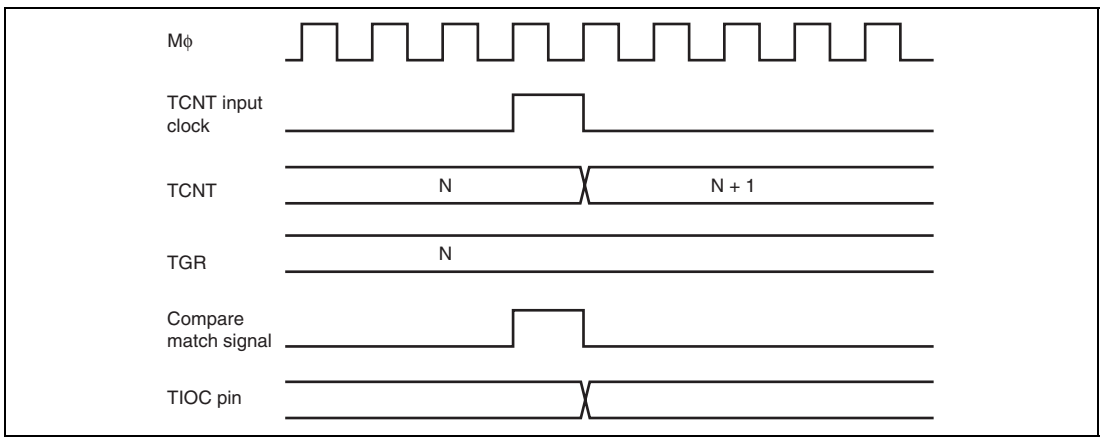


Figure 11.97 Output Compare Output Timing (Normal Mode/PWM Mode)

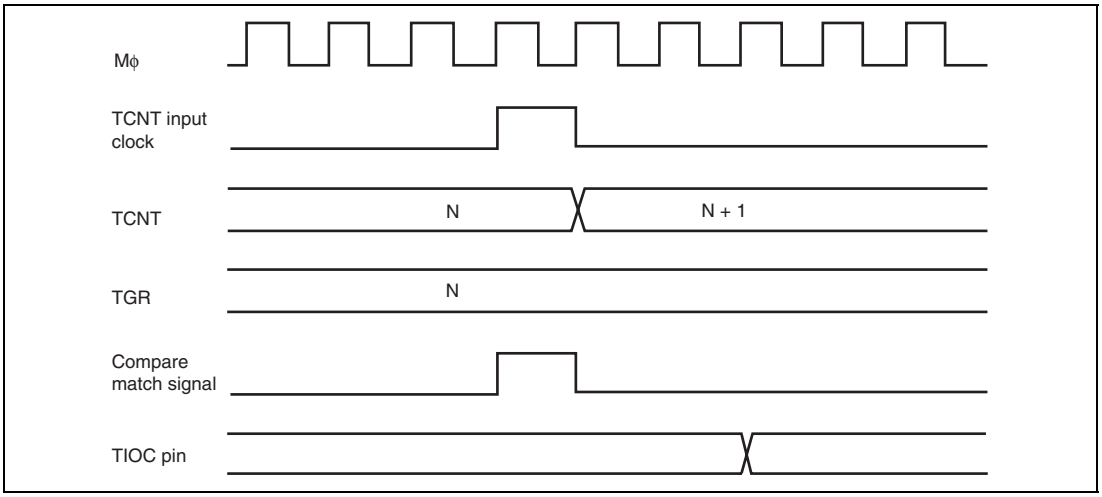


Figure 11.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 11.99 shows input capture signal timing.

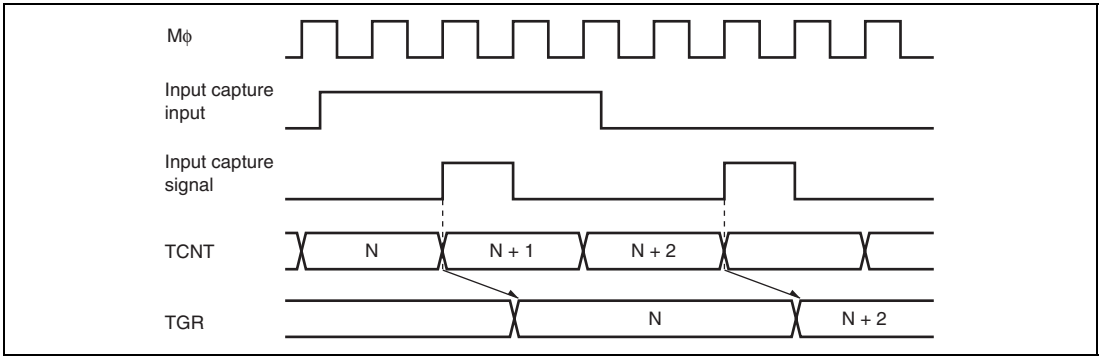


Figure 11.99 Input Capture Input Signal Timing

(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.

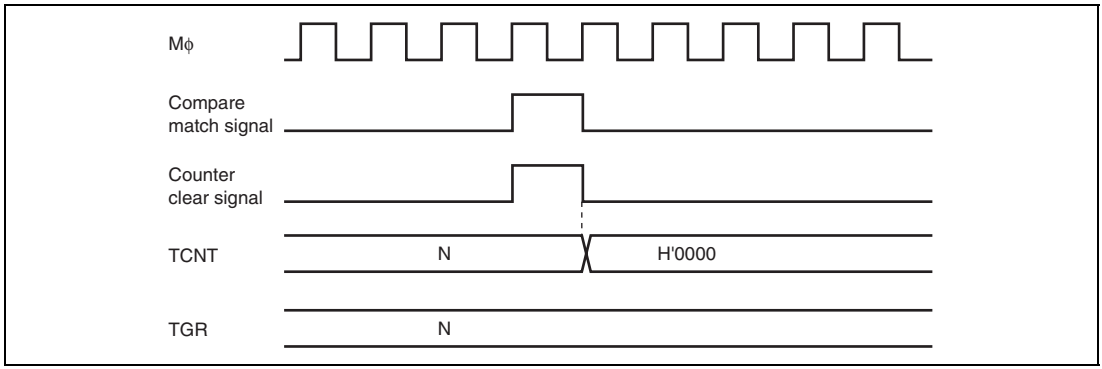


Figure 11.100 Counter Clear Timing (Compare Match) (Channels 0 to 4)

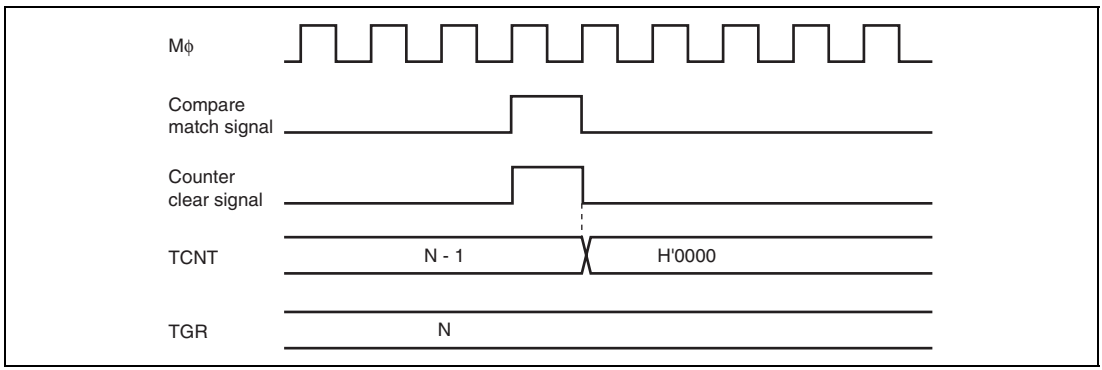


Figure 11.101 Counter Clear Timing (Compare Match) (Channel 5)

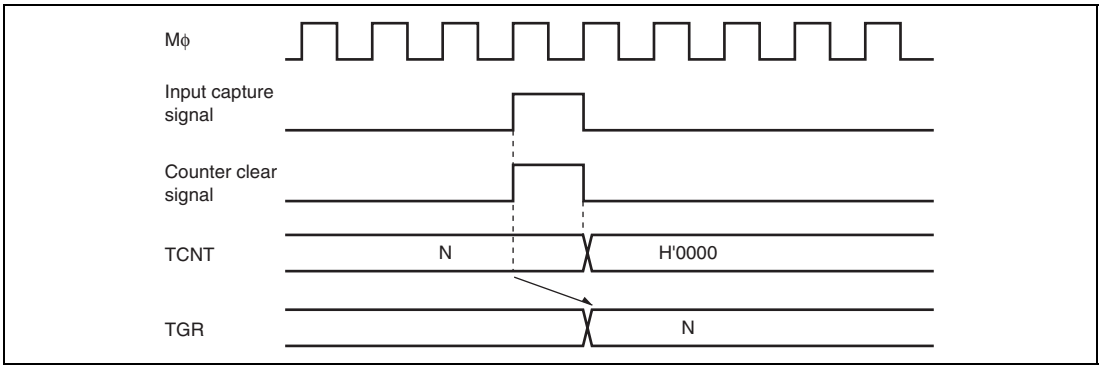


Figure 11.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)

(5) Buffer Operation Timing

Figures 11.103 to 11.105 show the timing in buffer operation.

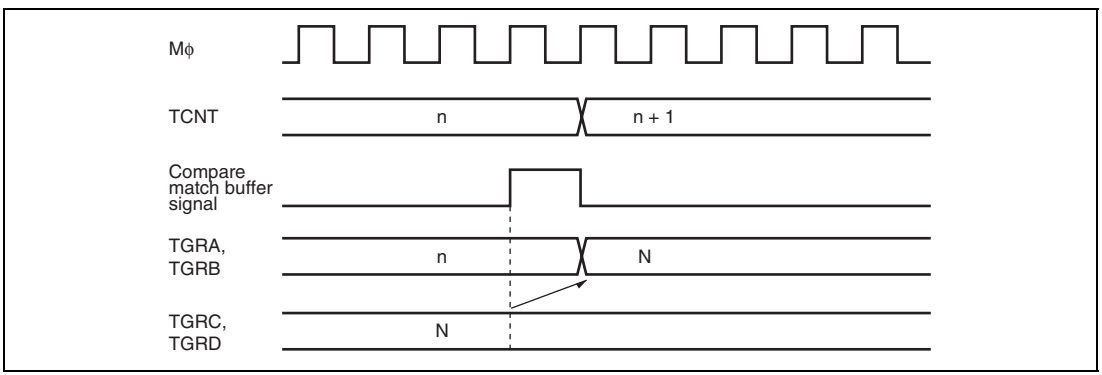


Figure 11.103 Buffer Operation Timing (Compare Match)

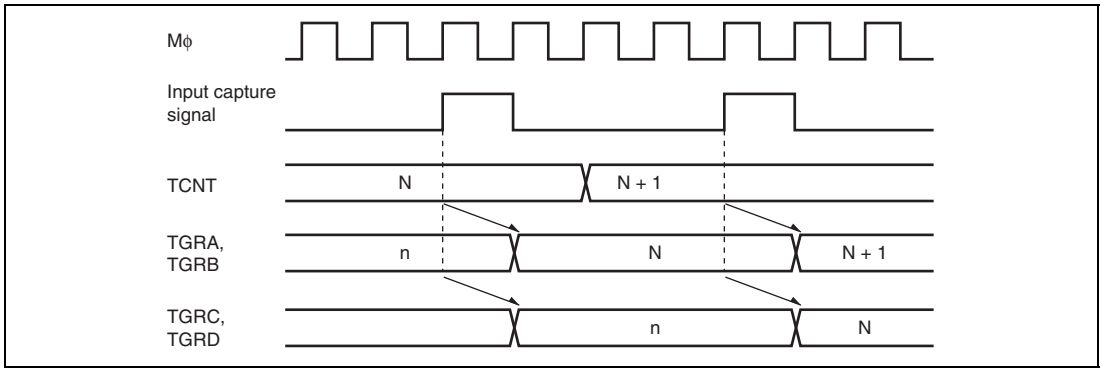


Figure 11.104 Buffer Operation Timing (Input Capture)

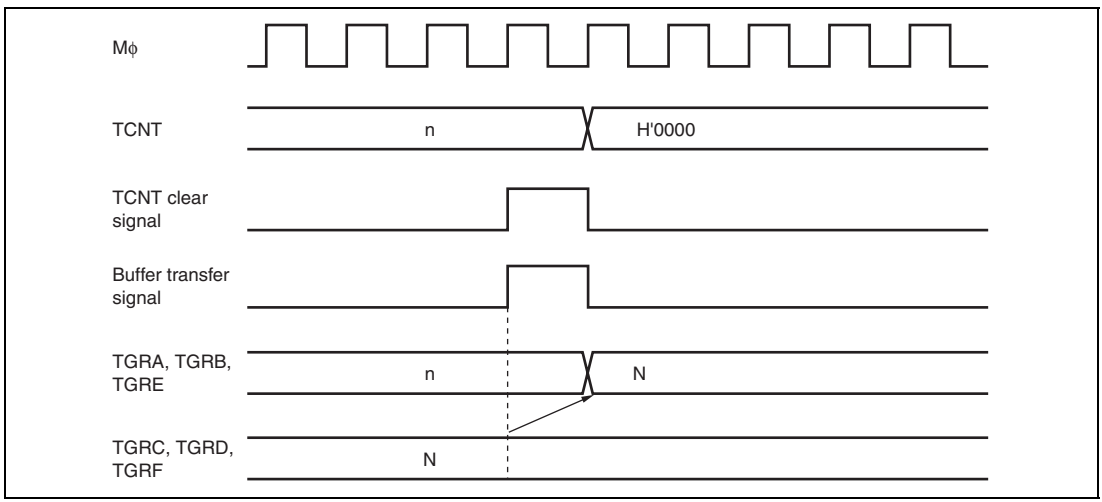


Figure 11.105 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 11.106 to 11.108 show the buffer transfer timing in complementary PWM mode.

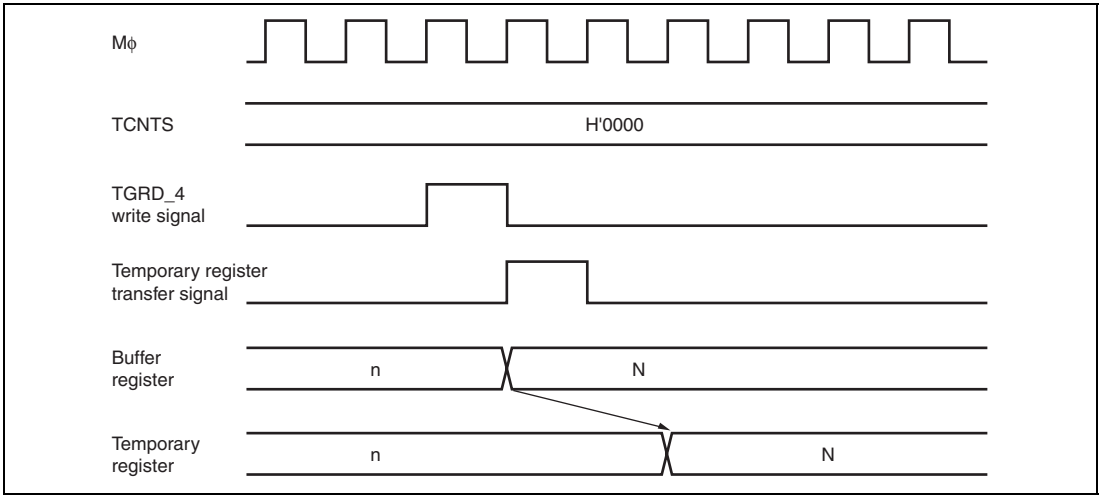


Figure 11.106 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

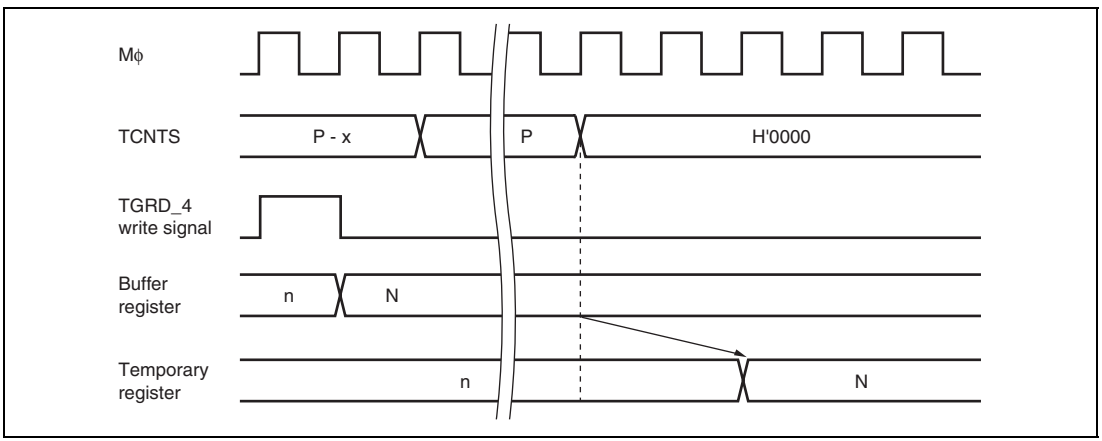


Figure 11.107 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

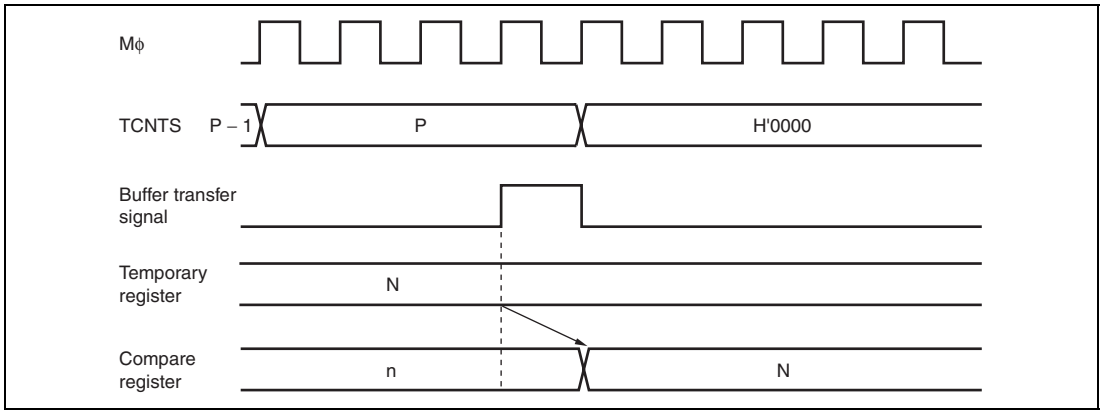


Figure 11.108 Transfer Timing from Temporary Register to Compare Register

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.

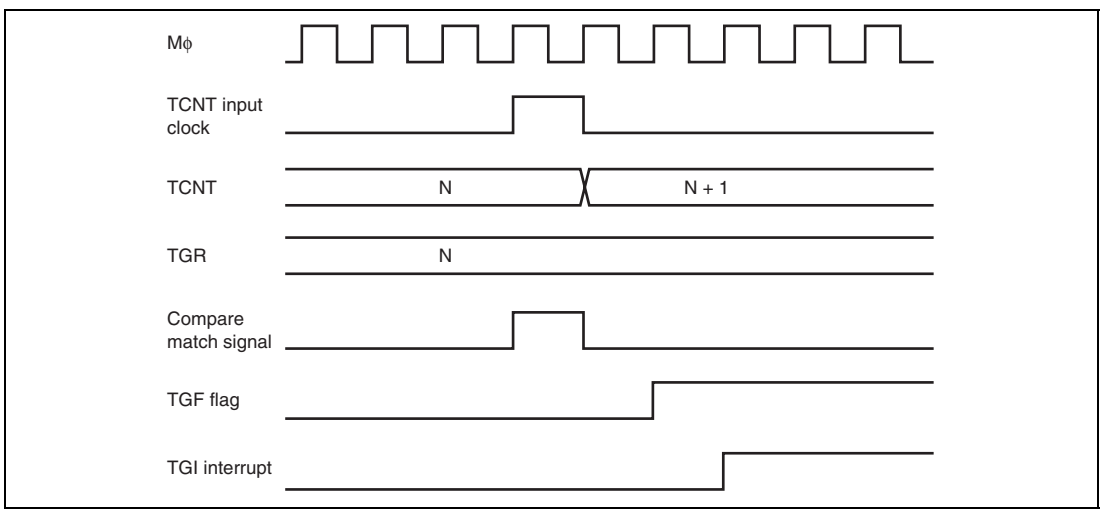


Figure 11.109 TGI Interrupt Timing (Compare Match)

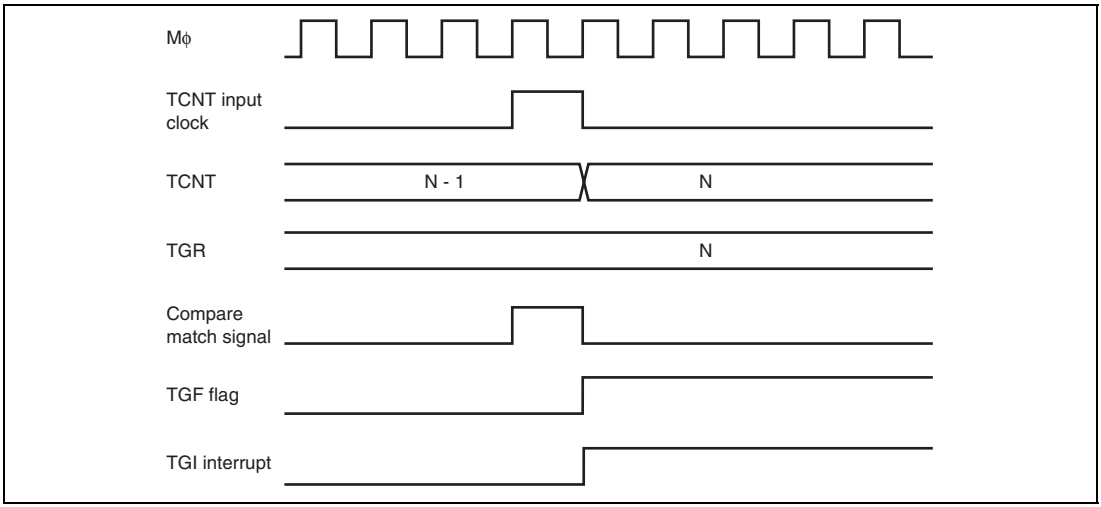


Figure 11.110 TGI Interrupt Timing (Compare Match) (Channel 5)

(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.

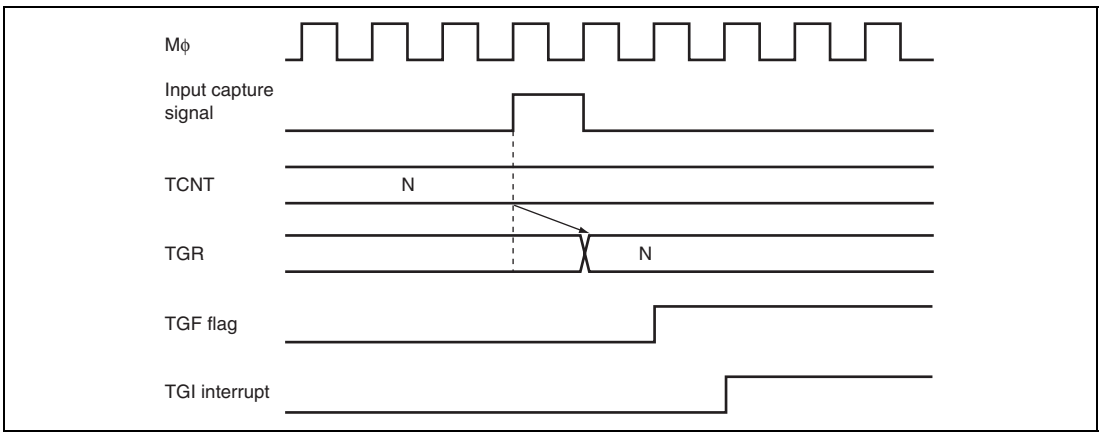


Figure 11.111 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)

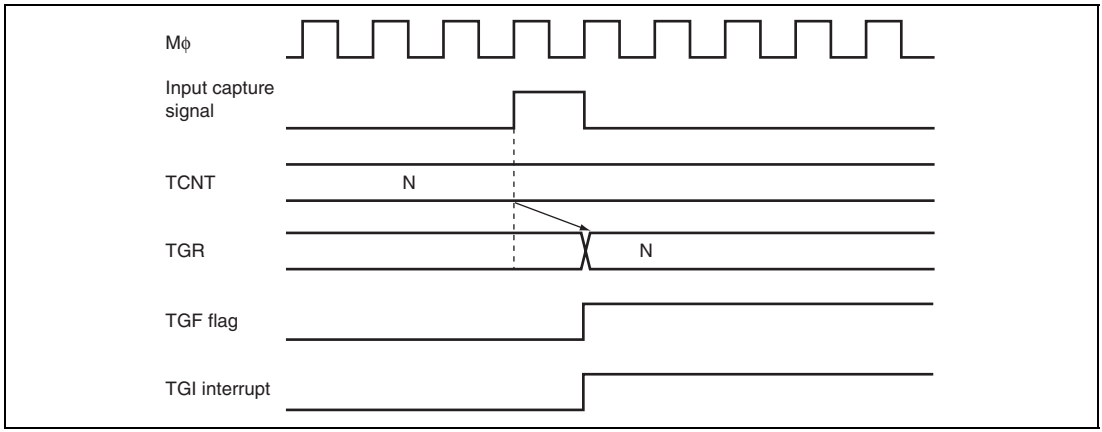


Figure 11.112 TGI Interrupt Timing (Input Capture) (Channel 5)

(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.

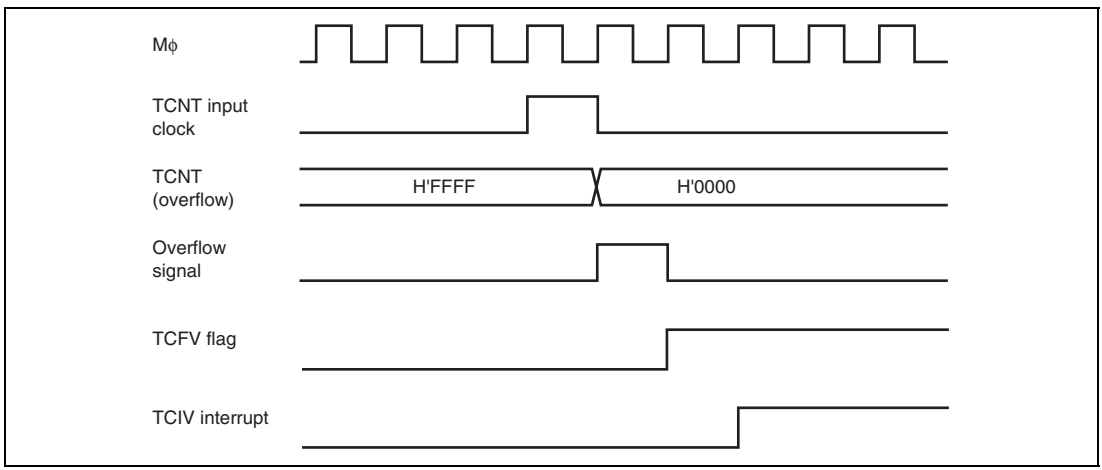


Figure 11.113 TCIV Interrupt Setting Timing

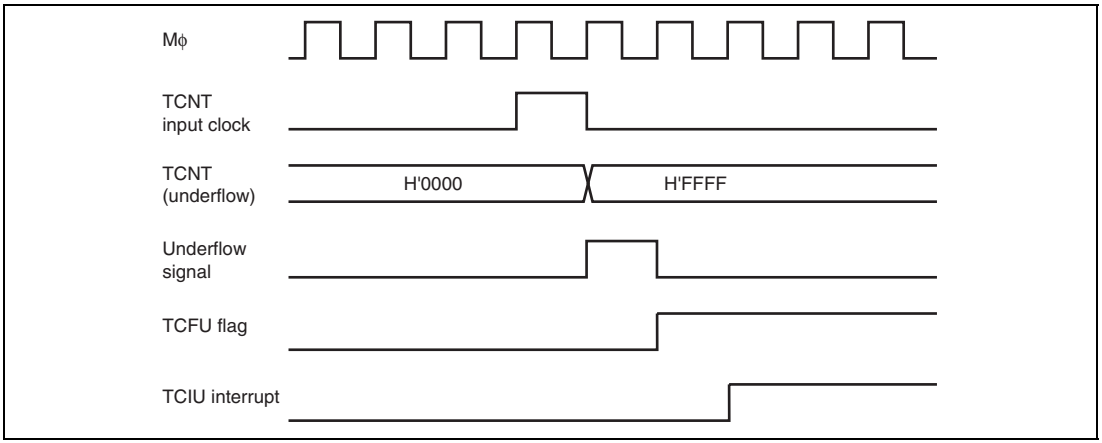


Figure 11.114 TCIU Interrupt Setting Timing

(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.

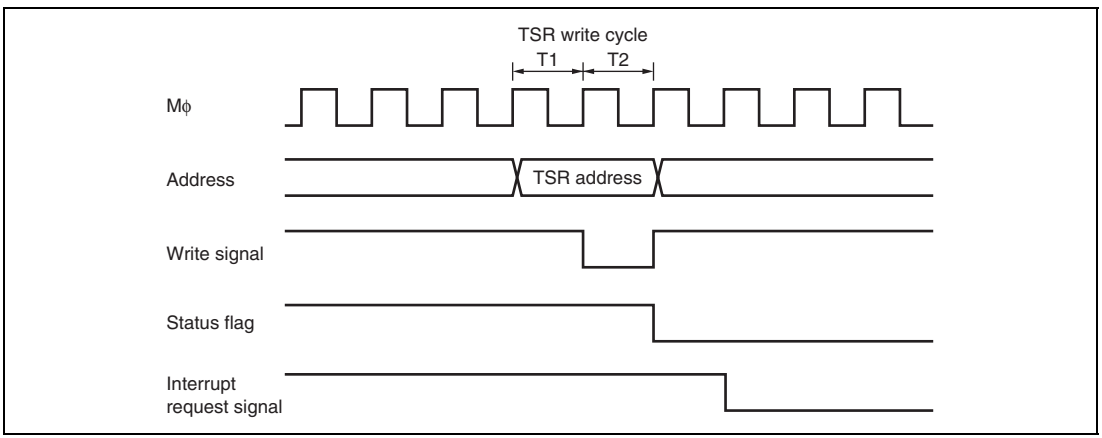


Figure 11.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)

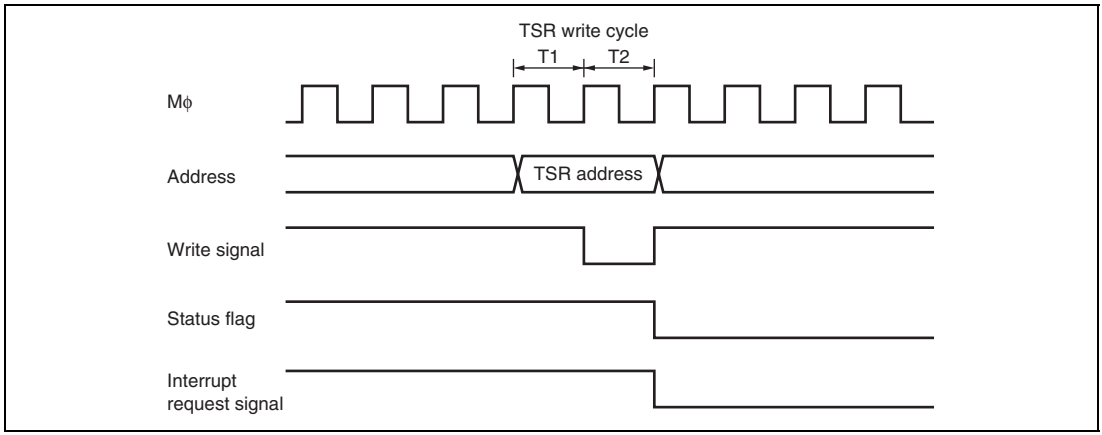


Figure 11.116 Timing for Status Flag Clearing by CPU (Channel 5)

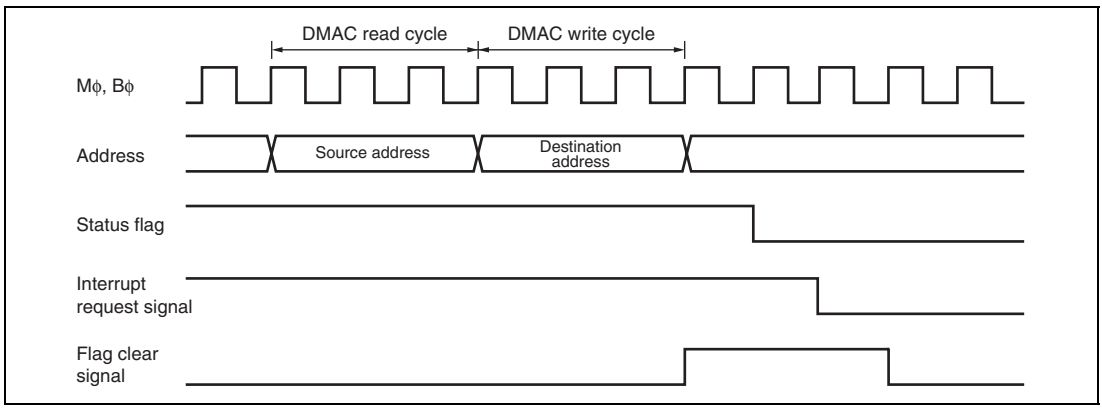


Figure 11.117 Timing for Status Flag Clearing by DTC Activation (Channels 0 to 4)

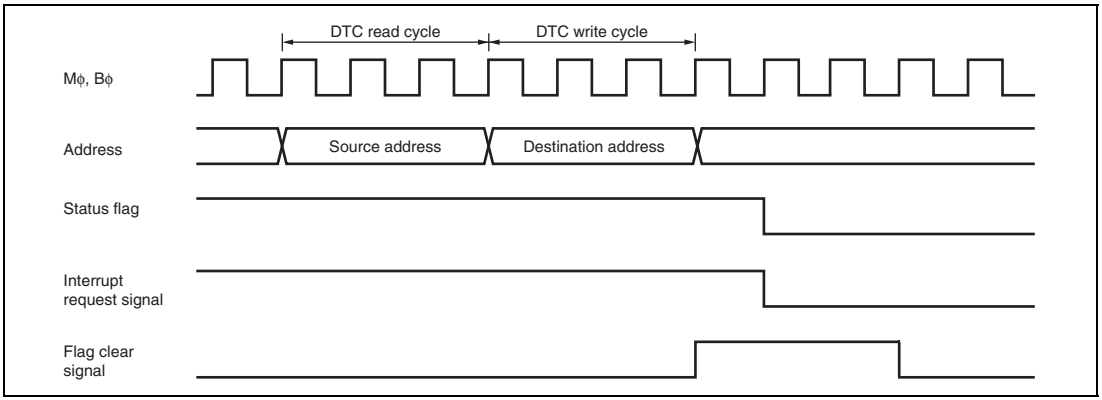


Figure 11.118 Timing for Status Flag Clearing by DTC Activation (Channel 5)

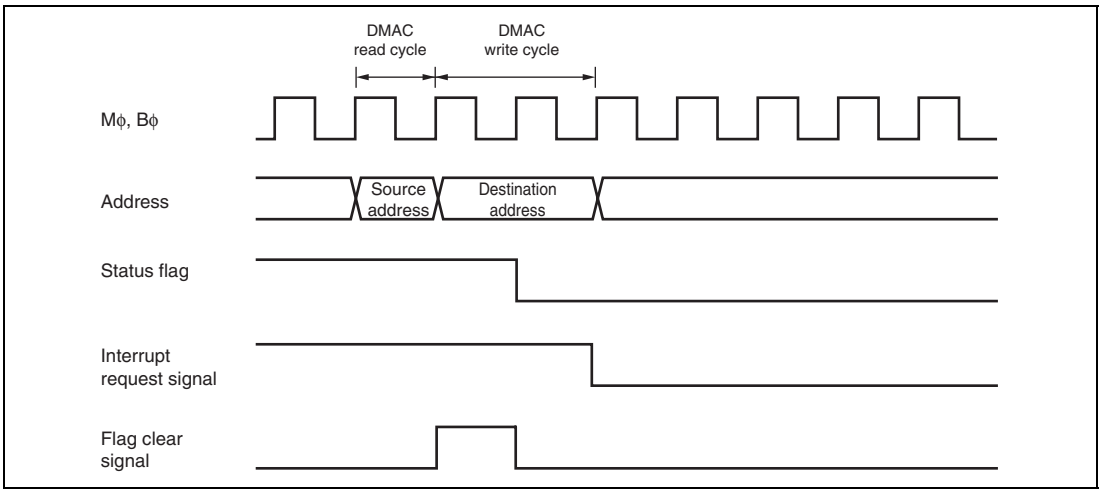


Figure 11.119 Timing for Status Flag Clearing by DMAC Activation

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 26, 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.

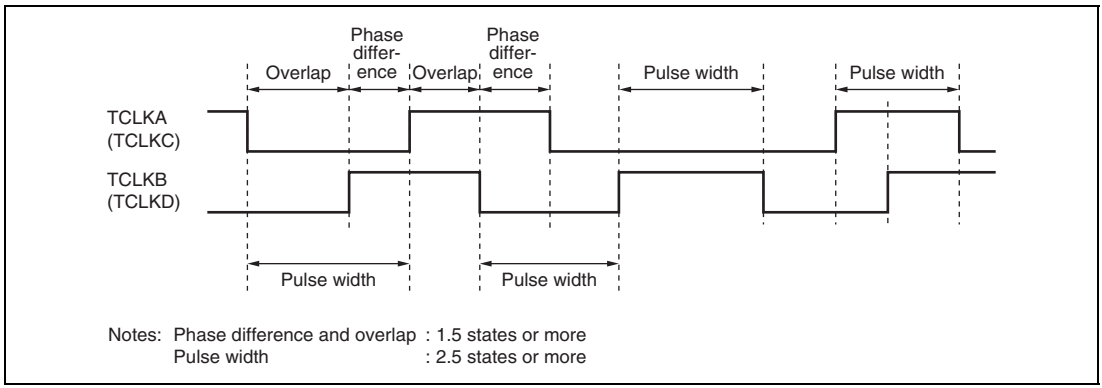


Figure 11.120 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

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

$$f = \frac{M\phi}{(N + 1)}$$

- Channel 5

$$f = \frac{M\phi}{N}$$

Where f: Counter frequency
Mϕ: 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.

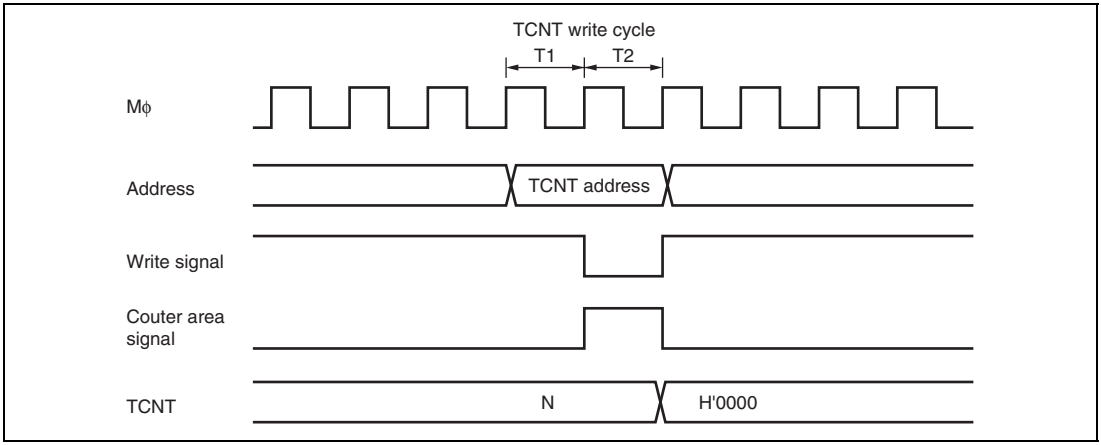


Figure 11.121 Contention between TCNT Write and Clear Operations

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.

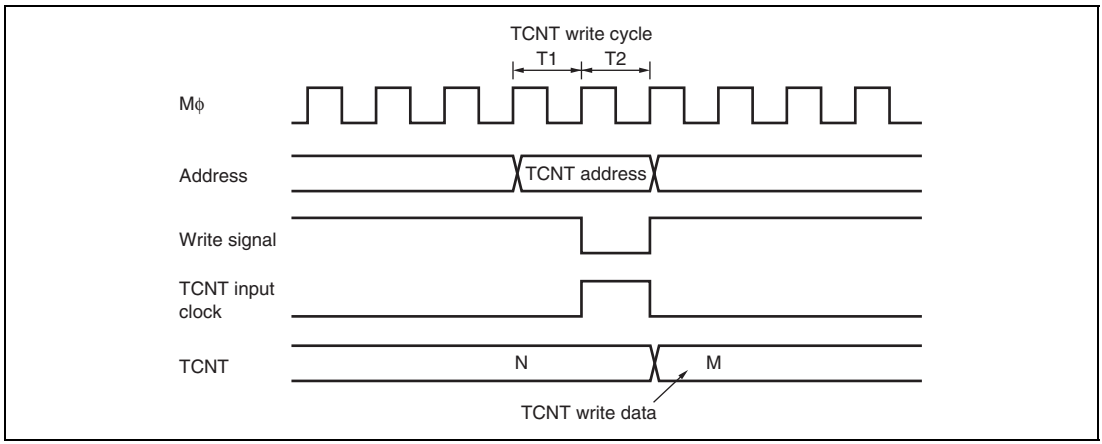


Figure 11.122 Contention between TCNT Write and Increment Operations

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.

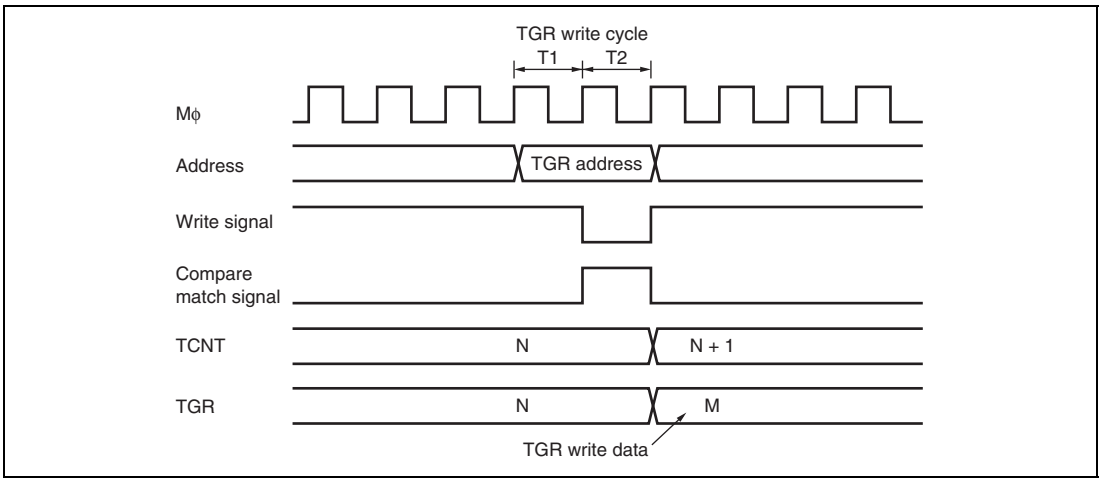


Figure 11.123 Contention between TGR Write and Compare Match

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.

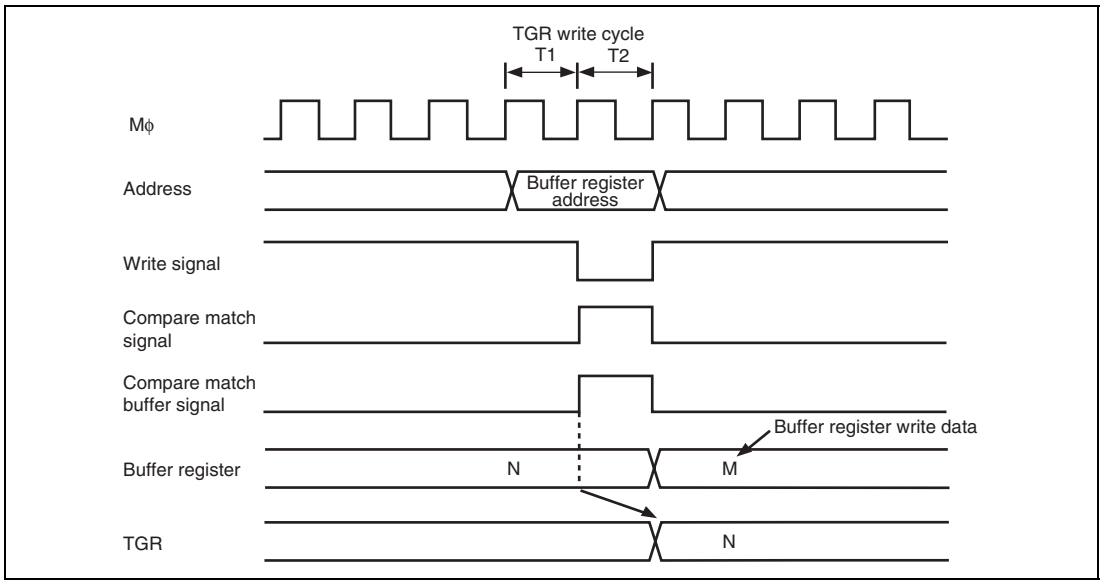


Figure 11.124 Contention between Buffer Register Write and Compare Match

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.

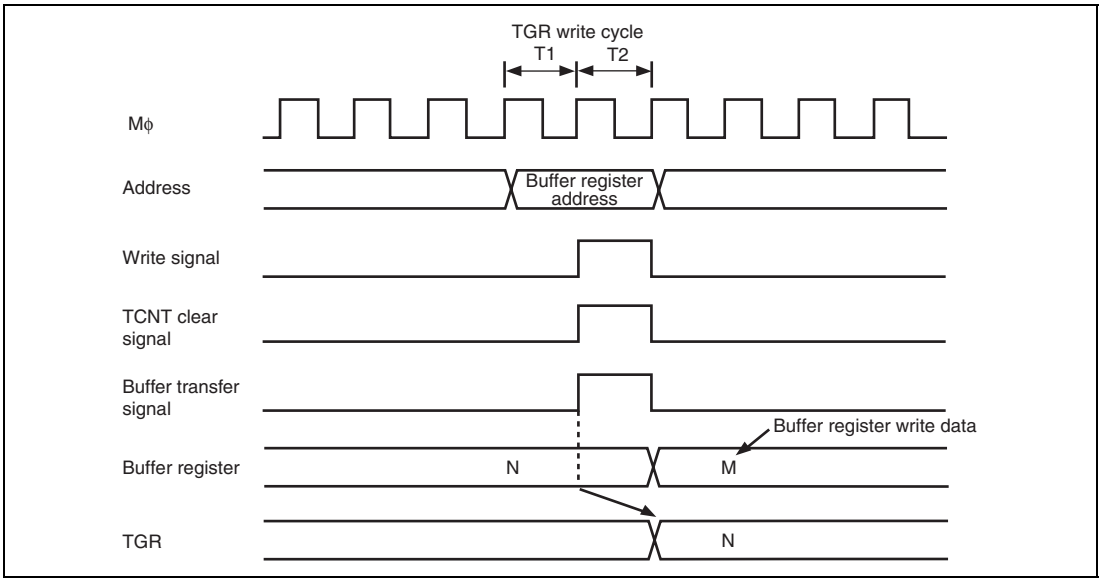


Figure 11.125 Contention between Buffer Register Write and TCNT Clear

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.

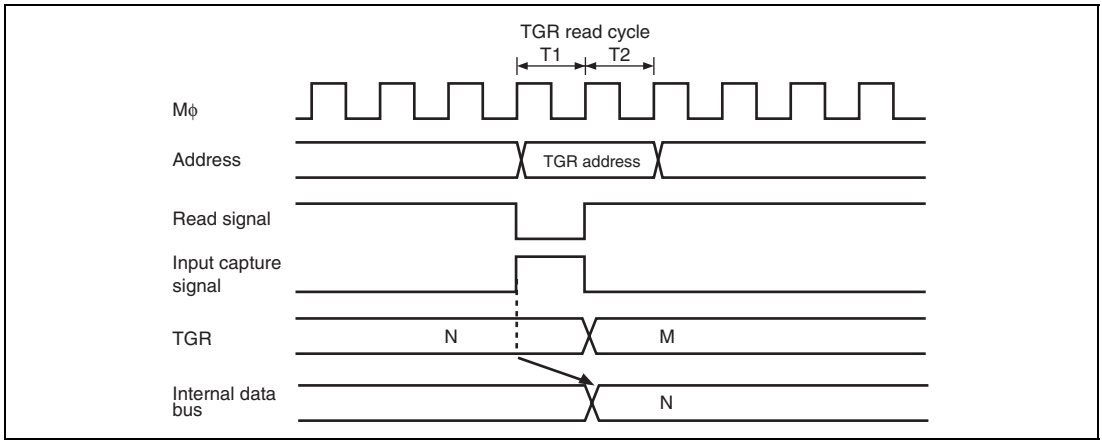


Figure 11.126 Contention between TGR Read and Input Capture (Channels 0 to 4)

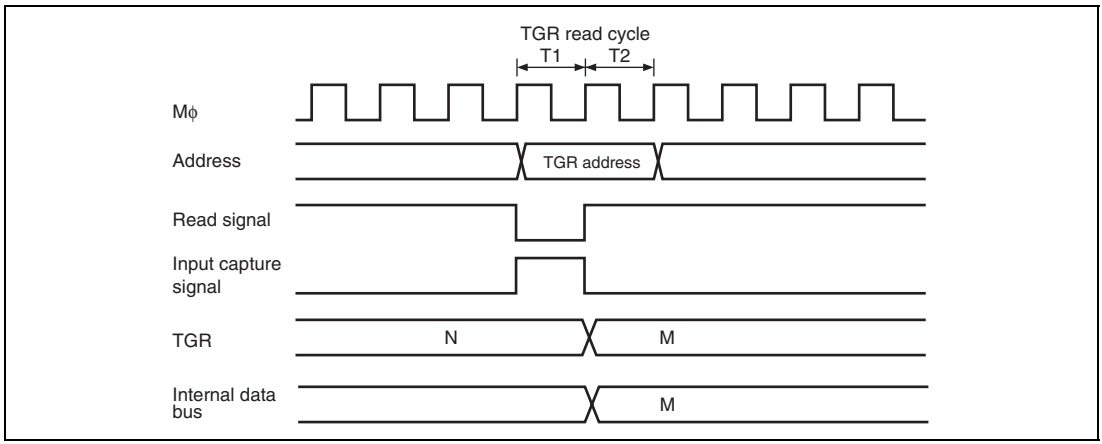


Figure 11.127 Contention between TGR Read and Input Capture (Channel 5)

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.

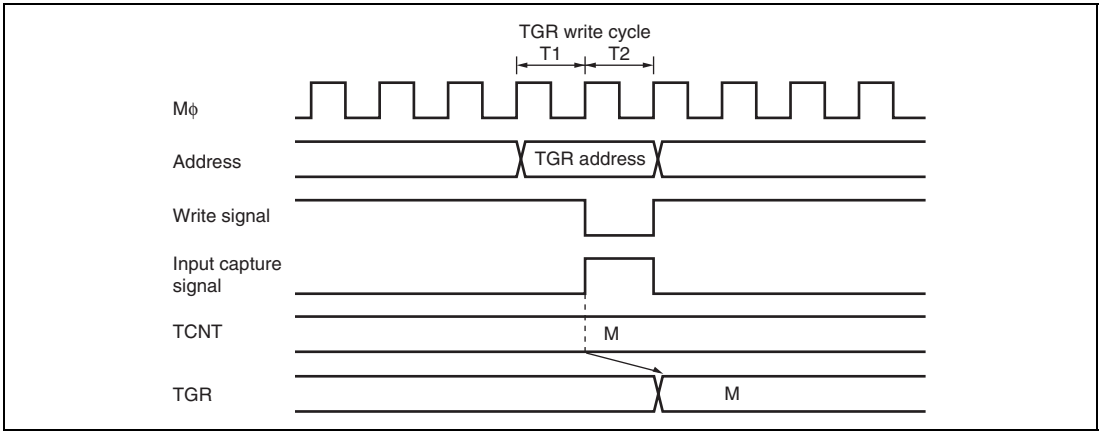


Figure 11.128 Contention between TGR Write and Input Capture (Channels 0 to 4)

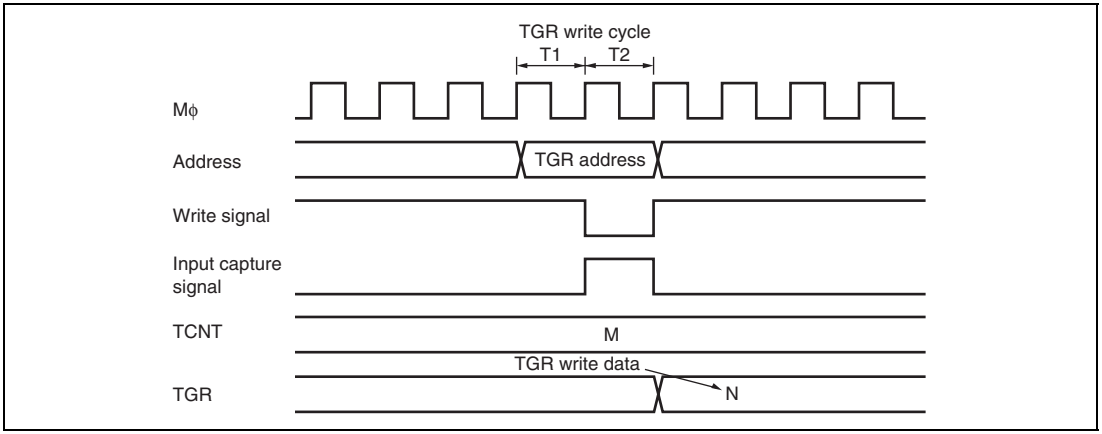


Figure 11.129 Contention between TGR Write and Input Capture (Channel 5)

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.

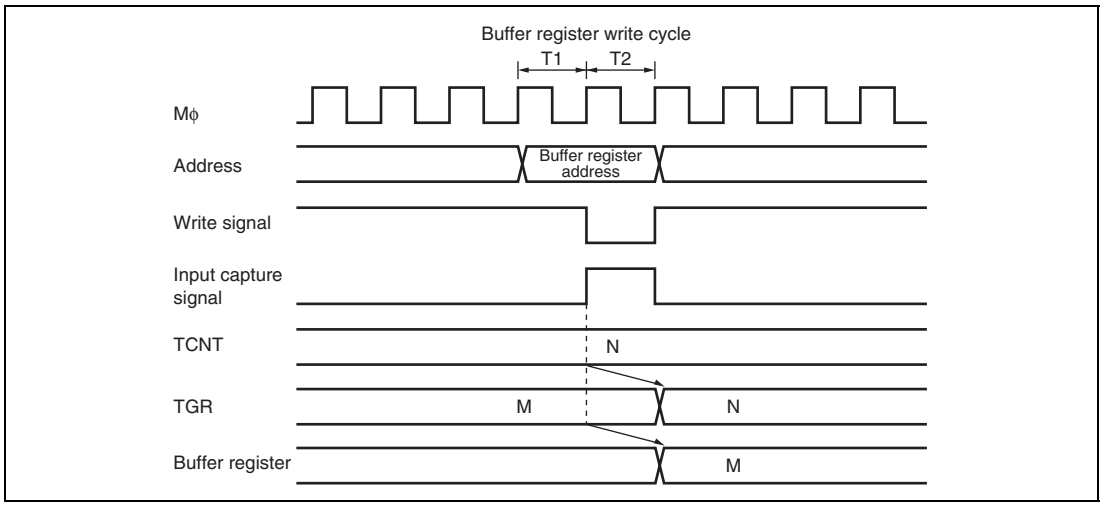


Figure 11.130 Contention between Buffer Register Write and Input Capture

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.

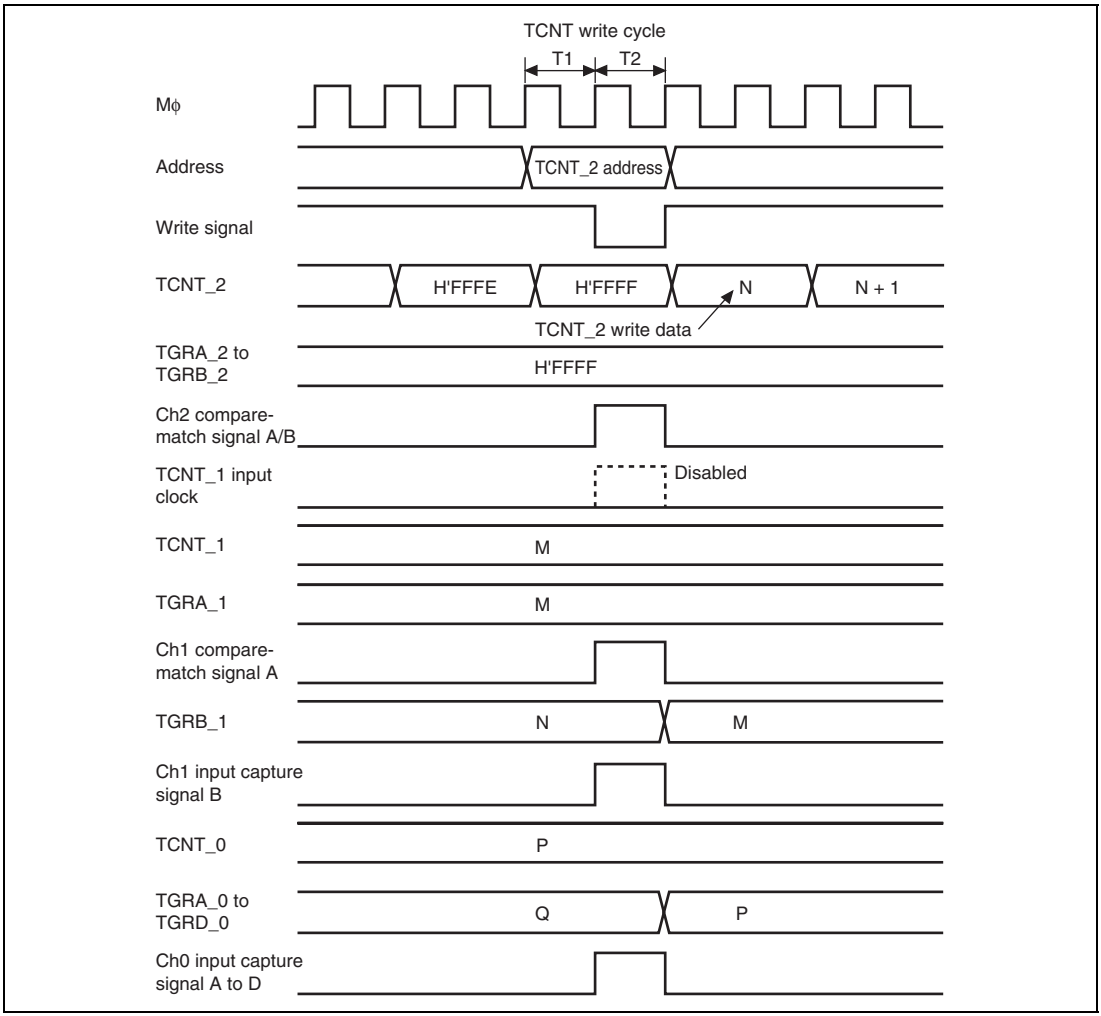


Figure 11.131 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

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.

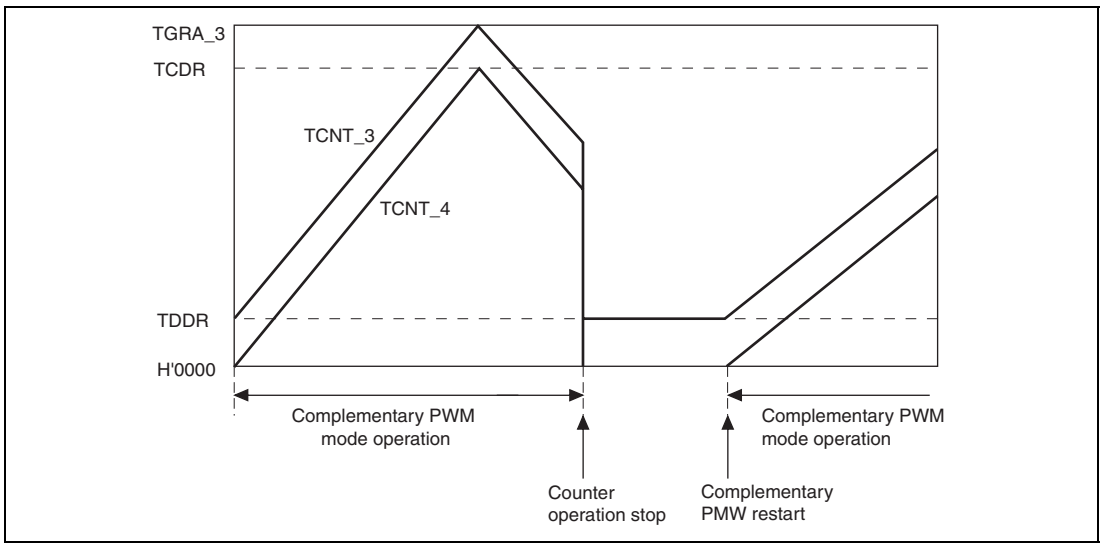


Figure 11.132 Counter Value during Complementary PWM Mode Stop

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.

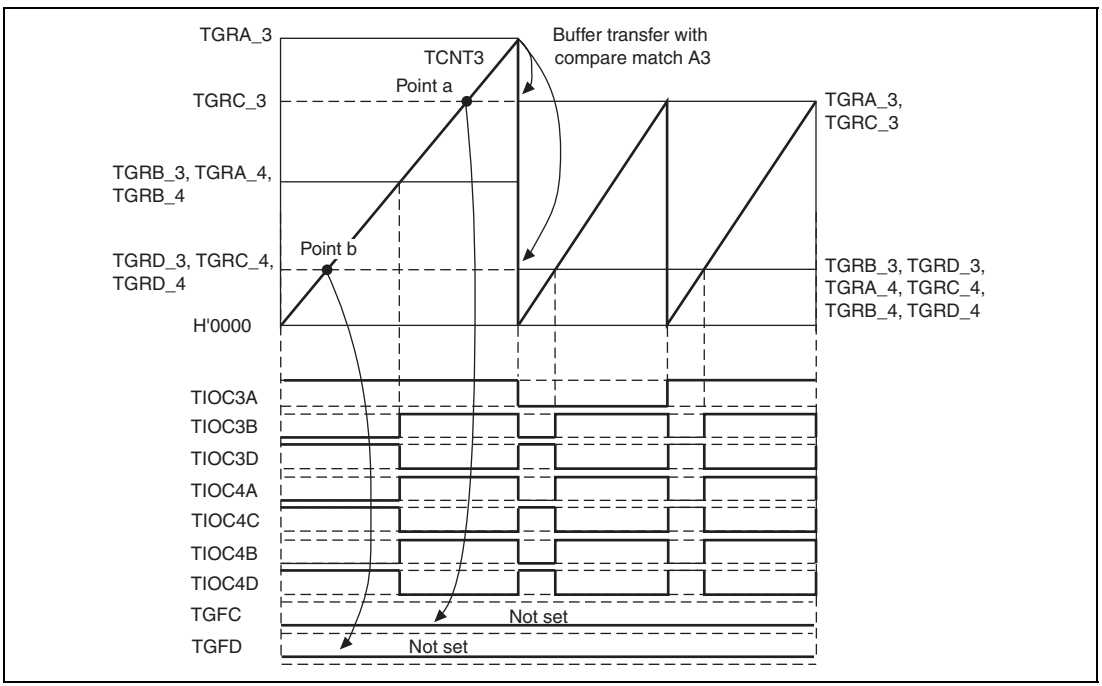


Figure 11.133 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

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.

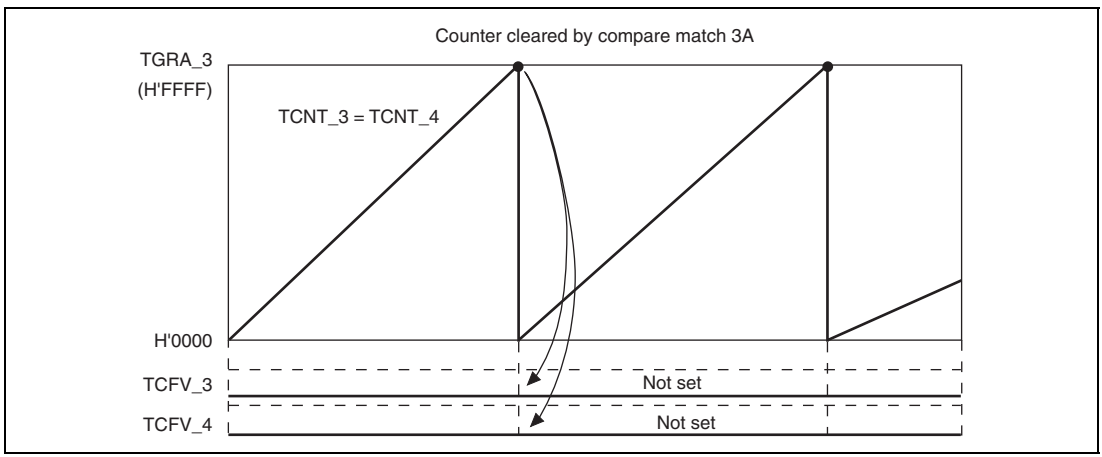


Figure 11.134 Reset Synchronous PWM Mode Overflow Flag

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.

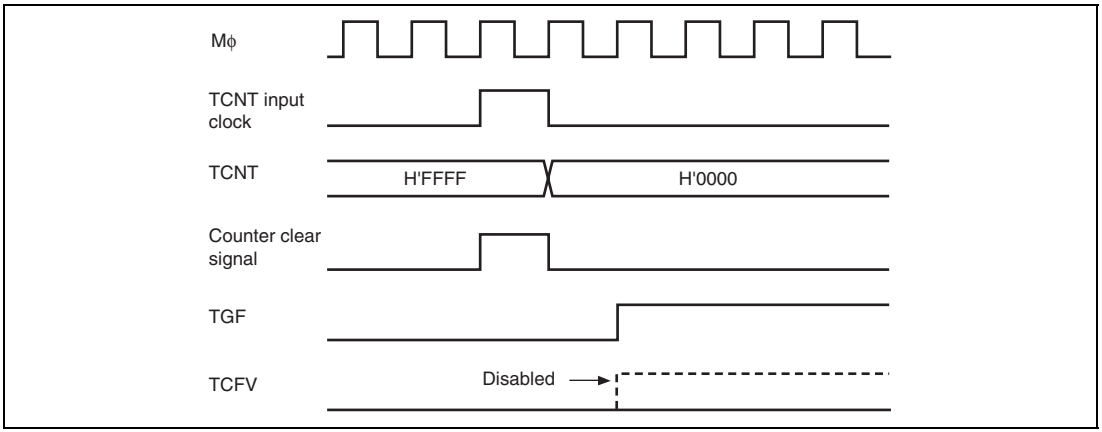


Figure 11.135 Contention between Overflow and Counter Clearing

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.

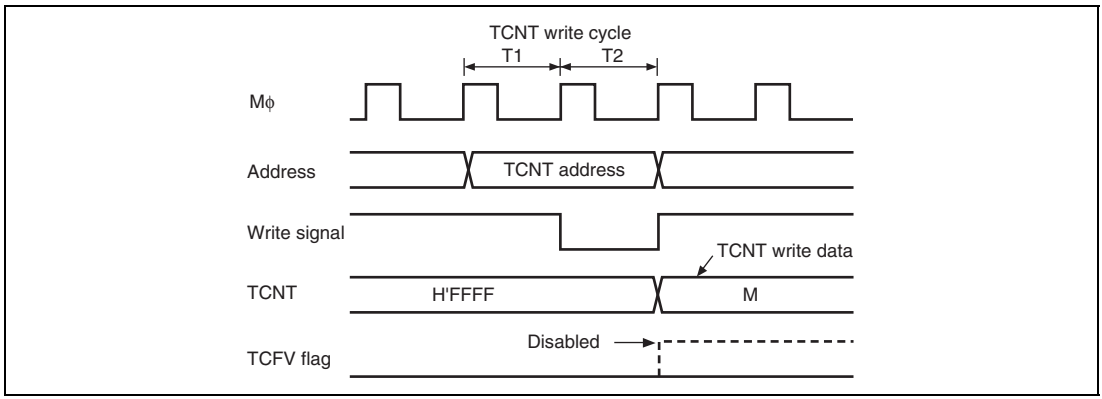


Figure 11.136 Contention between TCNT Write and Overflow

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).

11.8 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 11.137 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

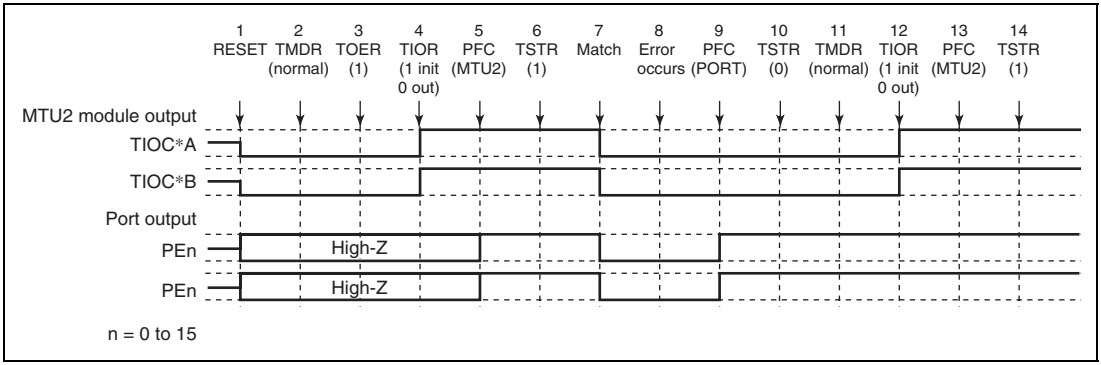


Figure 11.137 Error Occurrence in Normal Mode, Recovery in Normal Mode

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 11.138 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.

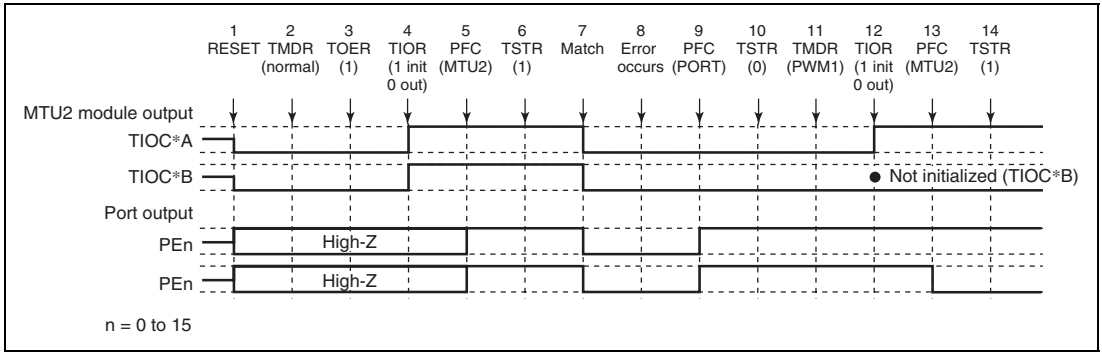


Figure 11.138 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.137.

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 11.139 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.

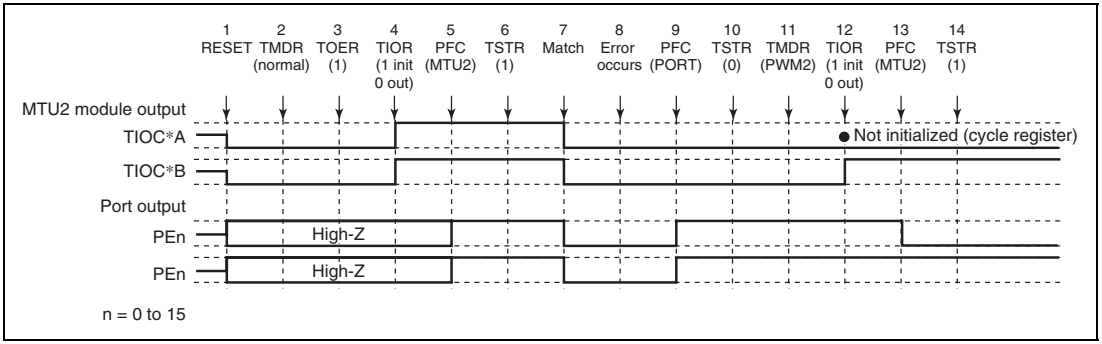


Figure 11.139 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.137.

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 11.140 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.

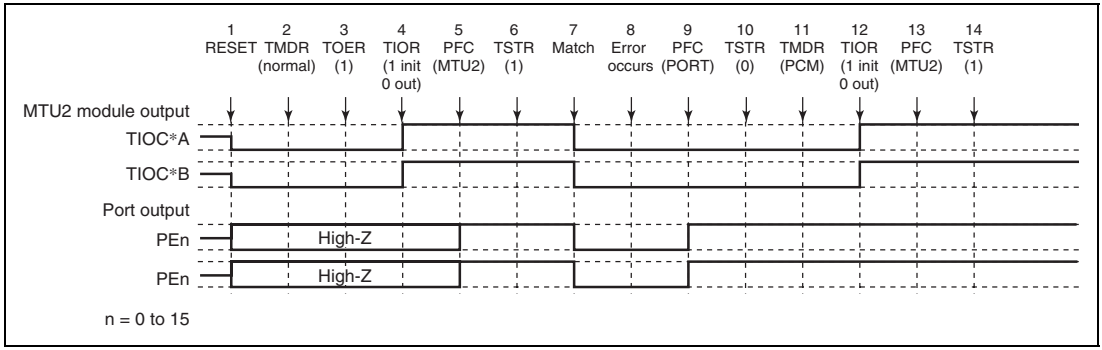


Figure 11.140 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.137.

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 11.141 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.

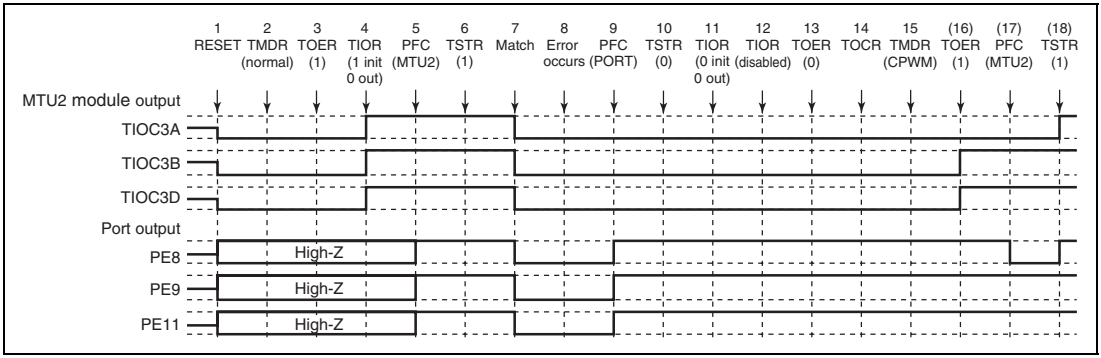


Figure 11.141 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.137.

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 11.142 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.

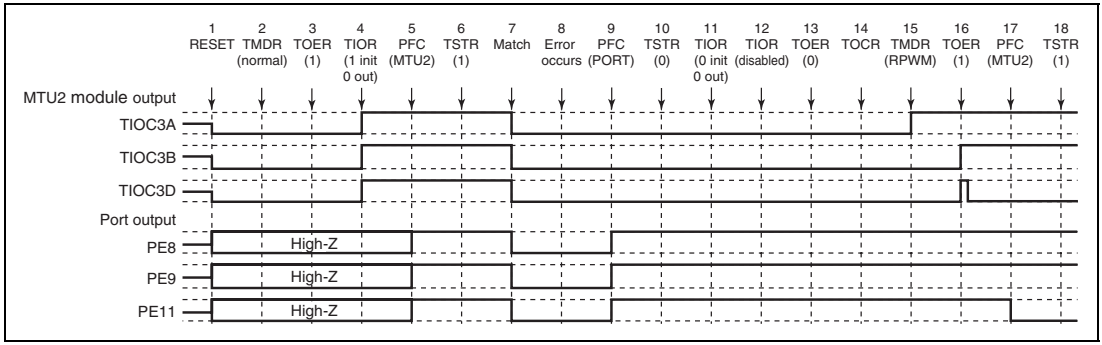


Figure 11.142 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 11.137.

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 11.143 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.

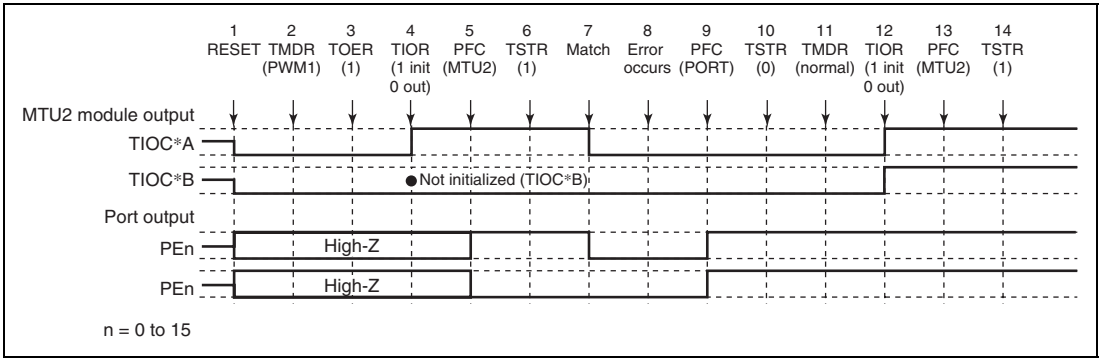


Figure 11.143 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

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 11.144 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.

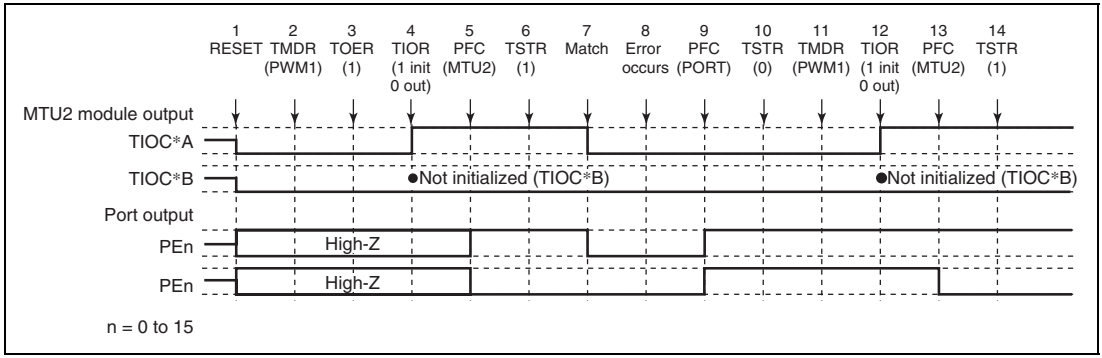


Figure 11.144 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.143.

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 11.145 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.

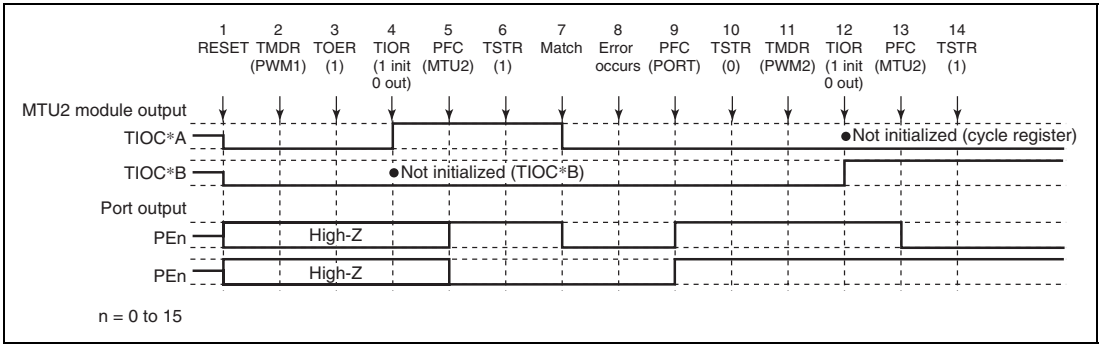


Figure 11.145 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 11.143.

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 11.146 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.

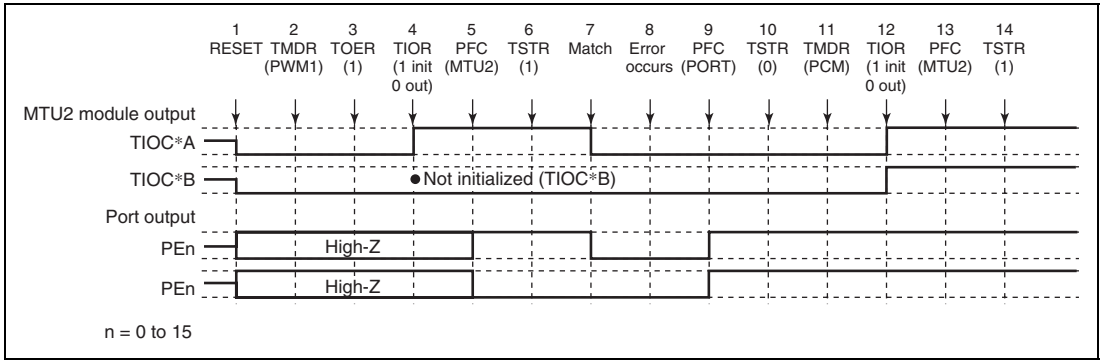


Figure 11.146 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 11.143.

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 11.147 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.

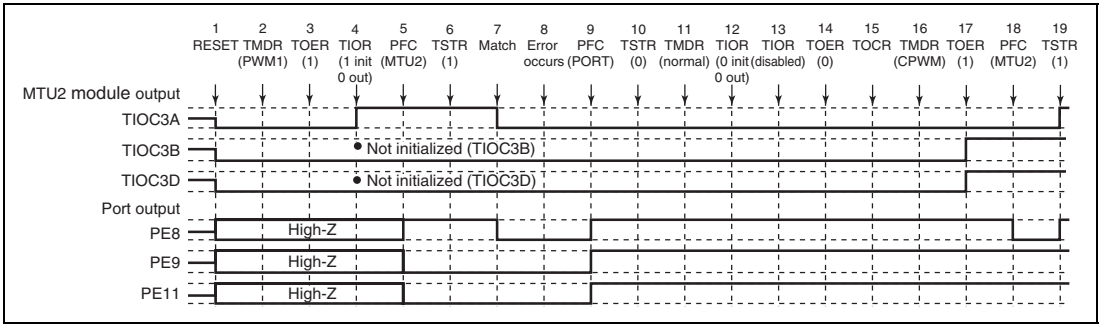


Figure 11.147 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.143.

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 11.148 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.

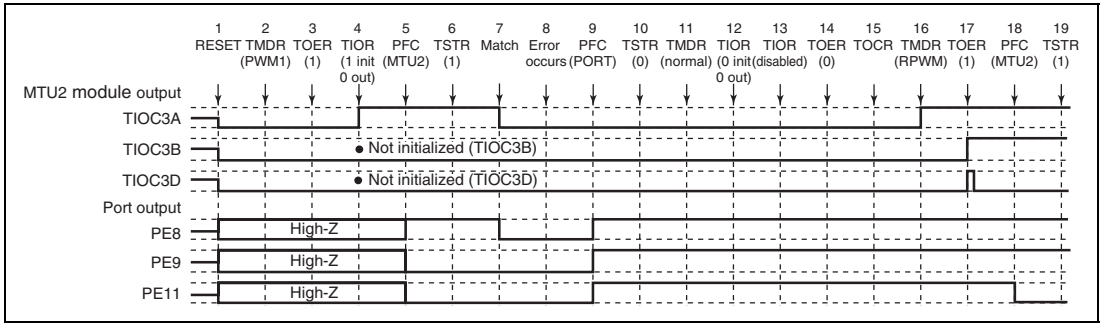


Figure 11.148 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 11.147.

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 11.149 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.

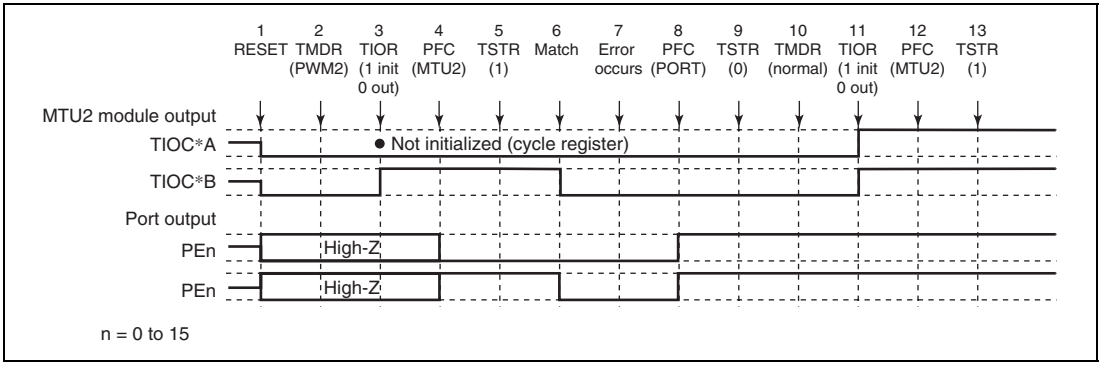


Figure 11.149 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

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 11.150 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.

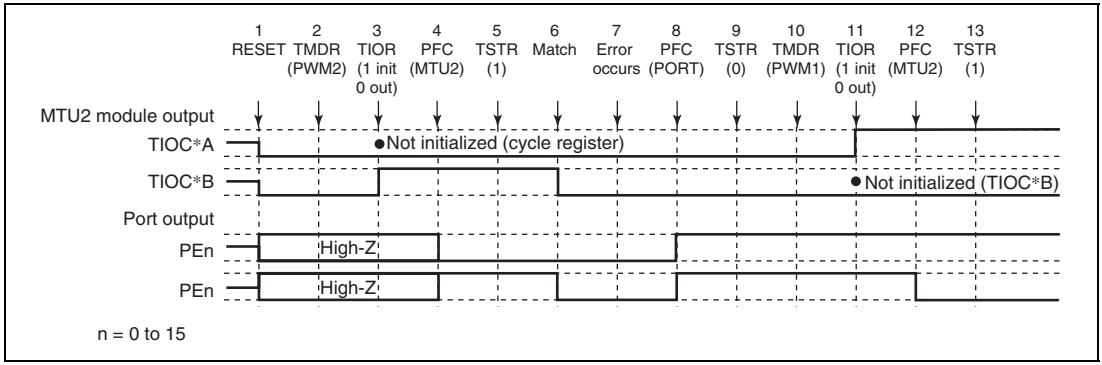


Figure 11.150 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.149.

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 11.151 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.

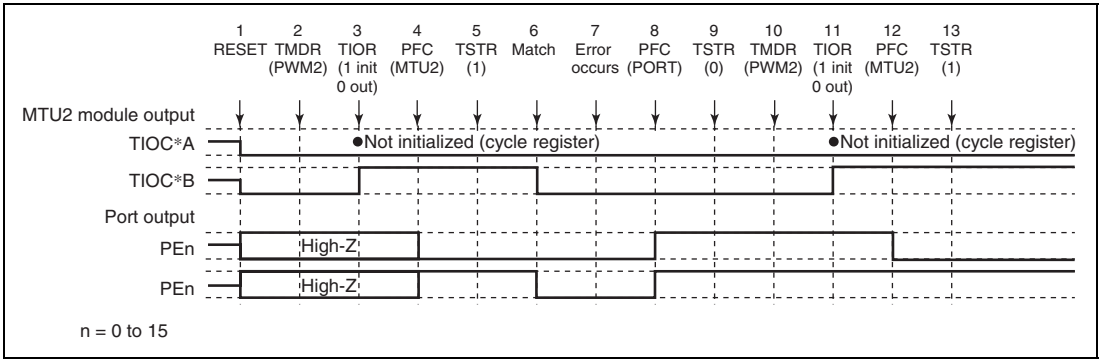


Figure 11.151 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.149.

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 11.152 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.

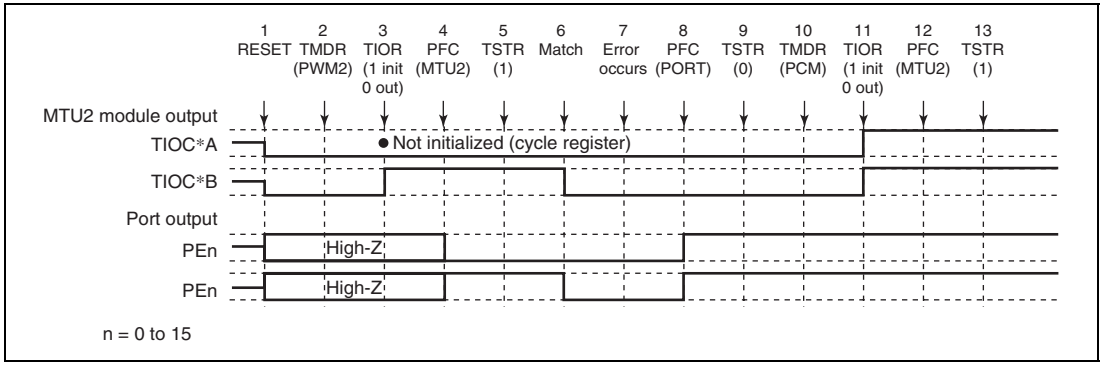


Figure 11.152 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.149.

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 11.153 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.

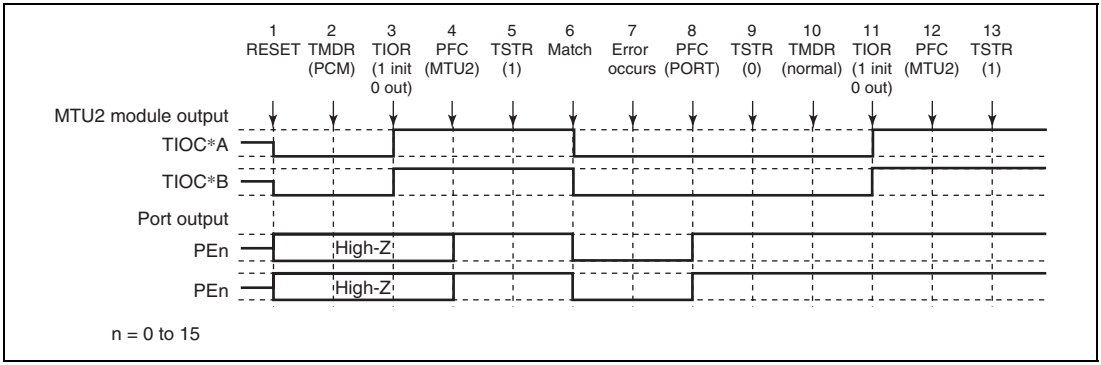


Figure 11.153 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

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 11.154 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.

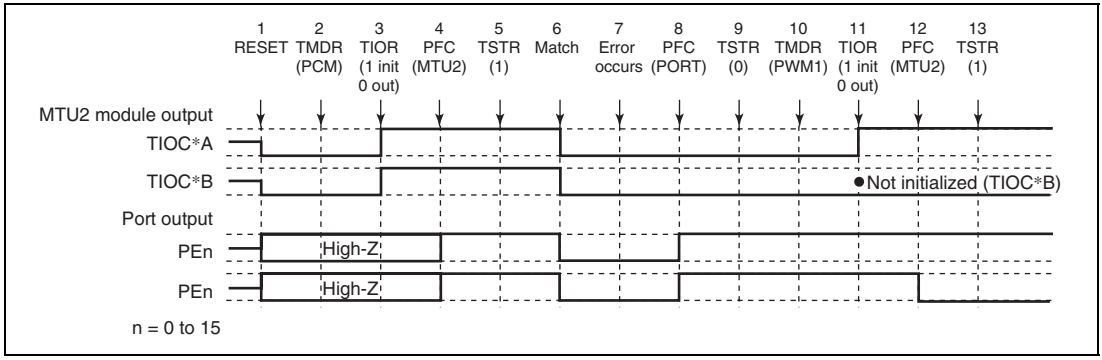


Figure 11.154 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 11.153.

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 11.155 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.

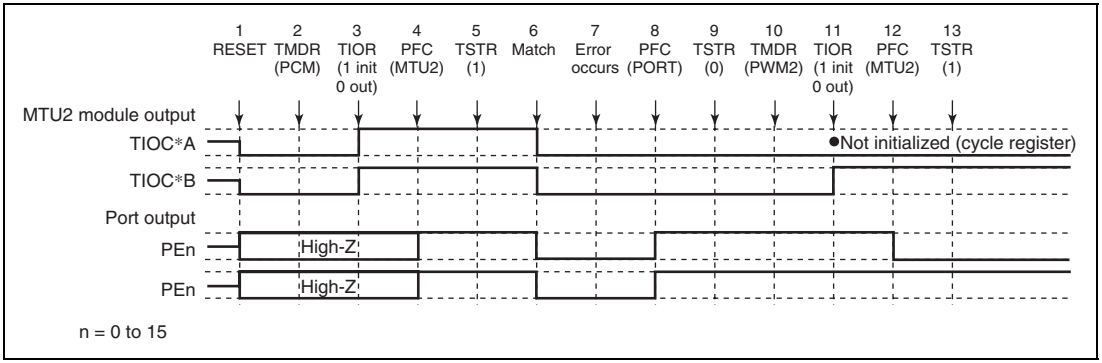


Figure 11.155 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.153.

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 11.156 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.

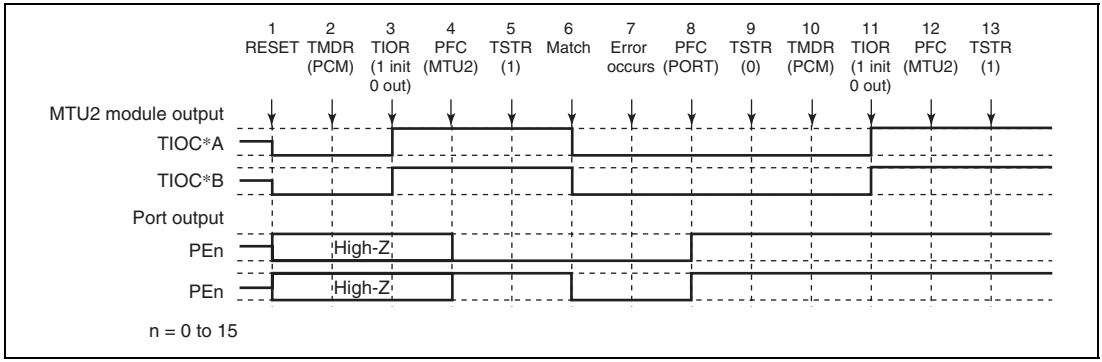


Figure 11.156 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 11.153.

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 11.157 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.

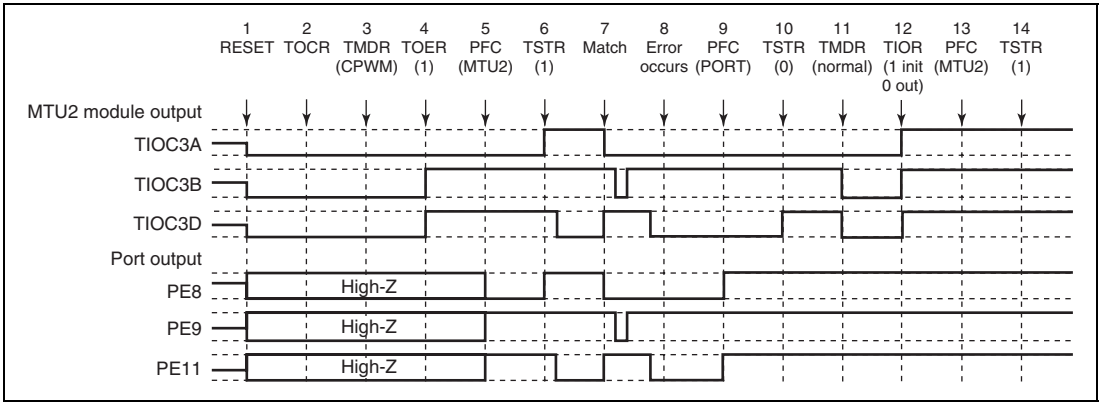


Figure 11.157 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

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 11.158 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.

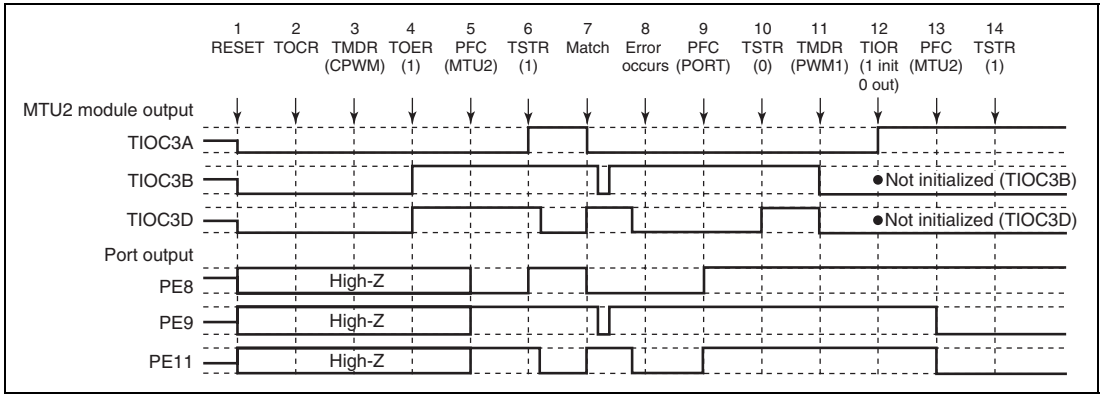


Figure 11.158 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.157.

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 11.159 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).

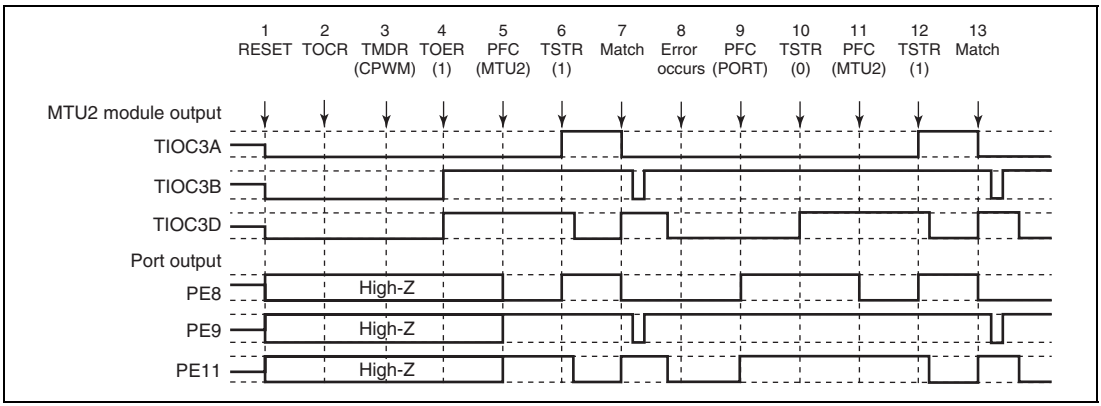


Figure 11.159 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.157.

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 11.160 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).

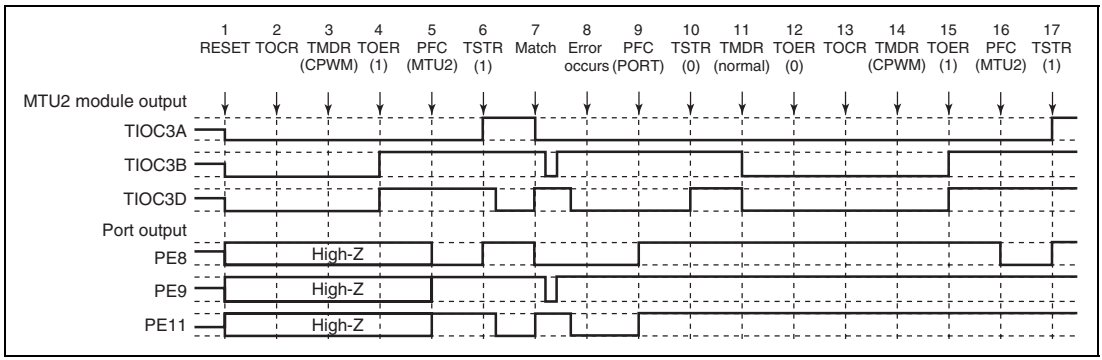


Figure 11.160 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.157.

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 11.161 shows an explanatory diagram of the case where an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode.

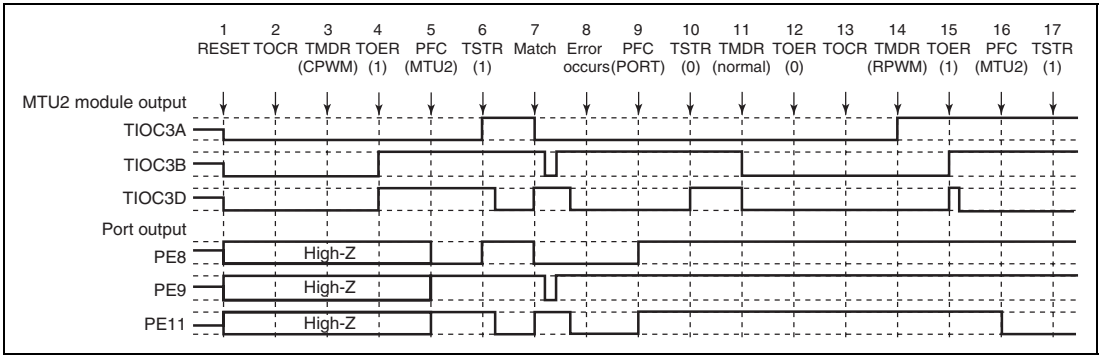


Figure 11.161 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.157.

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 11.162 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.

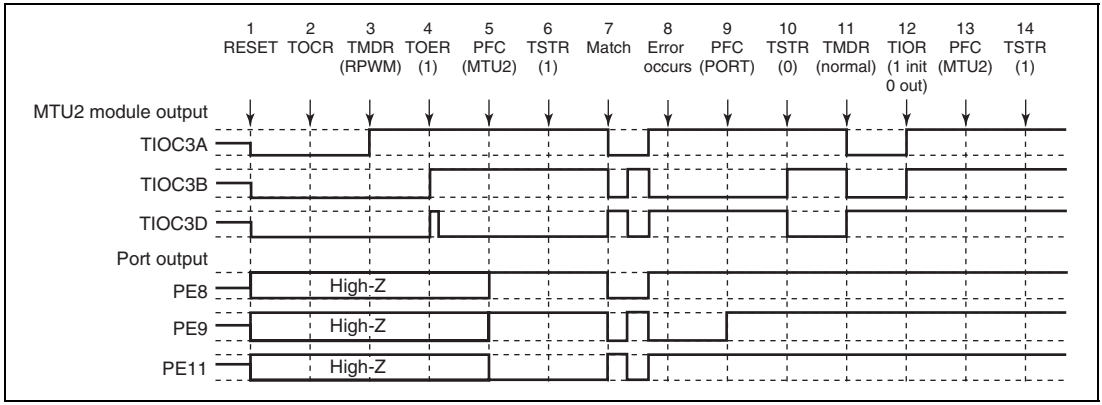


Figure 11.162 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

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 11.163 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.

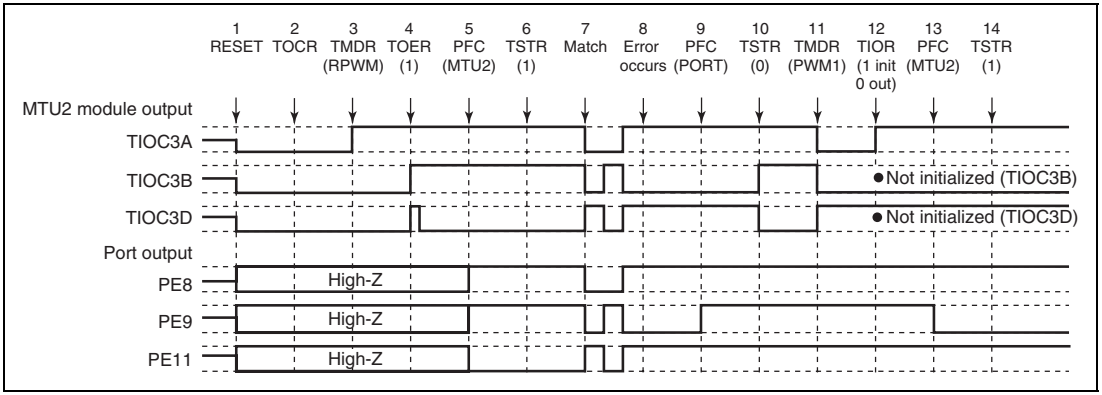


Figure 11.163 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 11.162.

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 11.164 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.

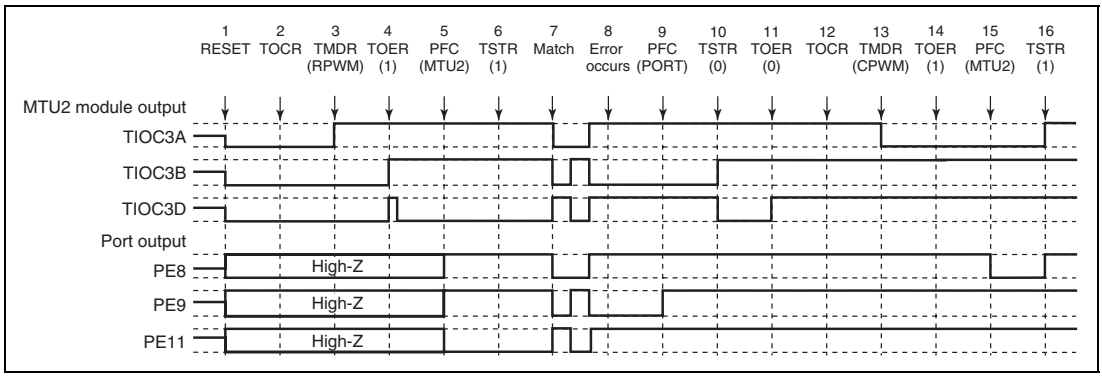


Figure 11.164 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 11.162.

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 11.165 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.

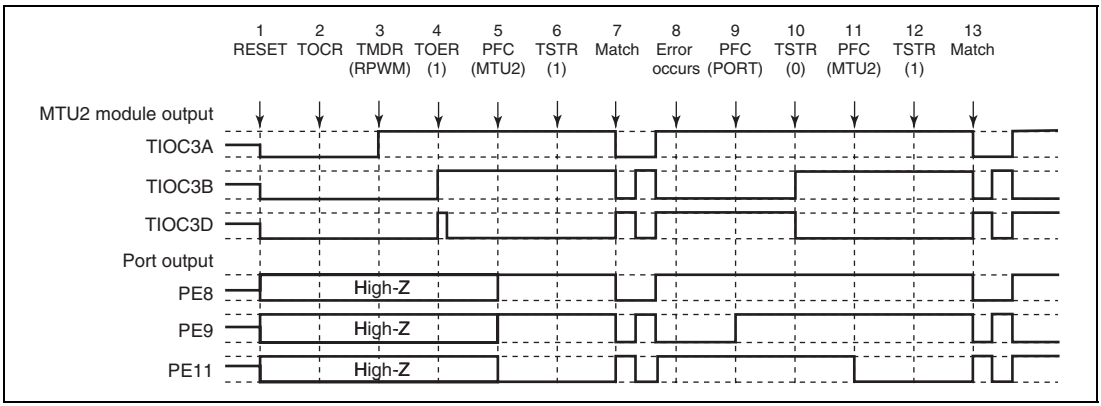


Figure 11.165 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 11.162.

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 operating frequency differs depending on whether the MTU2S is used for the complementary PWM mode output function or other functions as follows:

- Operating frequency for complementary PWM mode output function
  A maximum of 100 MHz: SH7239B and SH7237B
  A maximum of 80 MHz: SH7239A and SH7237A
- Operating frequency for other functions
  A maximum of 50 MHz: SH7239B and SH7237B
  A maximum of 40 MHz: SH7239A and SH7237A

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		√	√	—
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		—	—	√

Item	Channel 3	Channel 4	Channel 5
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
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 28, 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 and the pins for channel 0 of the MTU2 in high-impedance state, depending on the change on the $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ 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 $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ 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 $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins falling-edge or low-level sampling.
- Output pins to be placed in high-impedance state (complementary PWM pins and MTU2 CH0 pins) can be set separately (multiple pins can be set).
- 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.

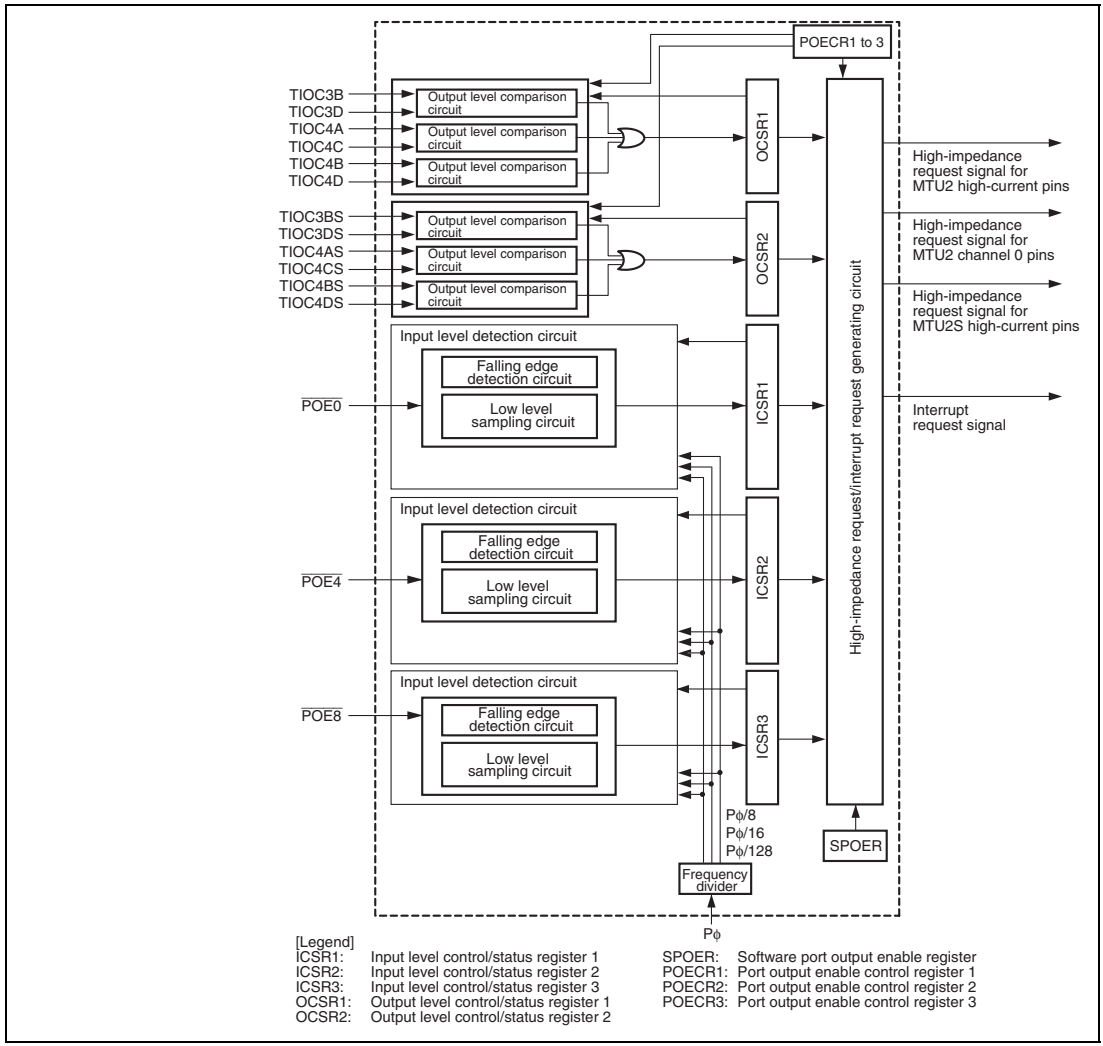


Figure 13.1 Block Diagram of POE2

13.2 Input/Output Pins

Table 13.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Port output enable input pins 0, 4, 8	$\overline{\text{POE0}}$, $\overline{\text{POE4}}$, $\overline{\text{POE8}}$	Input	Input request signals to place high-current pins for MTU2, MTU2S, and MTU2_CH0 in high-impedance state.

Table 13.2 shows output-level comparisons with pin combinations.

Table 13.2 Pin Combinations

Pin Combination	I/O	Description
TIOC3B and TIOC3D TIOC4A and TIOC4C TIOC4B and 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.
TIOC3BS and TIOC3DS TIOC4AS and TIOC4CS TIOC4BS and 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 or software 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
Port output enable control register 3	POECR3	R/W	H'00	H'FFFE500E	8

13.3.1 Input Level Control/Status Register 1 (ICSR1)

ICSR1 is a 16-bit readable/writable register that selects the $\overline{\text{POE0}}$ 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
	-	-	-	POE0F	-	-	-	PIE1	-	-	-	-	-	-	POE0M[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	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.

Bit	Bit Name	Initial Value	R/W	Description
8	PIE1	0	R/W	Port Interrupt Enable 1 Enables or disables interrupt requests when the POE0F bit in ICSR1 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	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)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

Note: *1 Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
7	MTU2PB4ZE	0	R/(W)*1	MTU2PB4 High-Impedance Enable Specifies whether to place the PB4/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*2 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)*[1]	MTU2PB3 High-Impedance Enable Specifies whether to place the PB3/TIOC0C pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*[2] 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)*[1]	MTU2PB2 High-Impedance Enable Specifies whether to place the PB2/TIOC0B pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*[2] 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)*[1]	MTU2PB1 High-Impedance Enable Specifies whether to place the PB1/TIOC0A pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*[2] 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)*[1]	MTU2PE3 High-Impedance Enable Specifies whether to place the PE3/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*[2] 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)*[1]	MTU2PE2 High-Impedance Enable Specifies whether to place the PE2/TIOC0C pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*[2] 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
1	MTU2PE1ZE	0	R/(W)*1	MTU2PE1 High-Impedance Enable Specifies whether to place the PE1/TIOC0B pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*2 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
0	MTU2PE0ZE	0	R/(W)*1	MTU2PE0 High-Impedance Enable Specifies whether to place the PE0/TIOC0A pin for channel 0 in the MTU2 in high-impedance state when either the selected $\overline{\text{POE}}$ pin flag*2 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

Notes: *1 Can modified only once after a power-on reset.

*2 The POE8F flag is selected in the initial state. The $\overline{\text{POE0}}$ and $\overline{\text{POE4}}$ pins can also be controlled by setting POECR3.

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	-	-	-	-
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*1	R/(W)*1	R/(W)*1	R	R/(W)*1	R/(W)*1	R/(W)*1	R	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R

Note: *1 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)*1	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 either the selected $\overline{\text{POE}}$ pin flag*2 or 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
13	MTU2P2CZE	1	R/(W)*1	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 either the selected $\overline{\text{POE}}$ pin flag*2 or 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
12	MTU2P3CZE	1	R/(W)*1	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 either the selected $\overline{\text{POE}}$ pin flag*2 or 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)*1	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 either the selected $\overline{\text{POE}}$ pin flag*2 or 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)*[1]	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 either the selected $\overline{\text{POE}}$ pin flag*[2] or 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)*[1]	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 either the selected $\overline{\text{POE}}$ pin flag*[2] or 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.
6	MTU2SP4CZE	0	R/(W)*[1]	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 either the selected $\overline{\text{POE}}$ pin flag*[2] or 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
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 either the selected $\overline{\text{POE}}$ pin flag*2 or 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)*1	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 either the selected $\overline{\text{POE}}$ pin flag*2 or 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.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: *1 Can be modified only once after a power-on reset.

*2 The POE0F flag is selected in the initial state. The $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ pins can also be controlled by setting POECR3.

*3 The POE4F flag is selected in the initial state. The $\overline{\text{POE0}}$ and $\overline{\text{POE8}}$ pins can also be controlled by setting POECR3.

13.3.9 Port Output Enable Control Register 3 (POECR3)

POECR3 is an 8-bit readable/writable register that controls high-impedance state of the $\overline{\text{POE}}$ pins other than the pin selected by default.

Bit:	7	6	5	4	3	2	1	0
	-	-	IC2MTU2 CH0ZE	IC2MTU2 ZE	IC3MTU2 SZE	IC3MTU2 ZE	IC1MTU2 CH0ZE	IC1MTU2 SZE
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)*

Note: * Can be modified only once after a power-on reset.

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	IC2MTU2 CH0ZE	0	R/(W)*	IC2MTU2CH0 High-Impedance Enable Controls the high-impedance state of the high-current pins for MTU2 CH0 when the POE4F bit is set. 0: Disables the pins to be placed in the high-impedance state. 1: Enables the pins to be placed in the high-impedance state.
4	IC2MTU2ZE	0	R/(W)*	IC2MTU2 High-Impedance Enable Controls the high-impedance state of the high-current pins for MTU2 when the POE4F bit is set. 0: Disables the pins to be placed in the high-impedance state. 1: Enables the pins to be placed in the high-impedance state.

Bit	Bit Name	Initial Value	R/W	Description
3	IC3MTU2SZE	0	R/(W)*	IC3MTU2S High-Impedance Enable Controls the high-impedance state of the high-current pins for MTU2S when the POE8F bit is set. 0: Disables the pins to be placed in the high-impedance state. 1: Enables the pins to be placed in the high-impedance state.
2	IC3MTU2ZE	0	R/(W)*	IC3MTU2 High-Impedance Enable Controls the high-impedance state of the high-current pins for MTU2 when the POE8F bit is set. 0: Disables the pins to be placed in the high-impedance state. 1: Enables the pins to be placed in the high-impedance state.
1	IC1MTU2 CH0ZE	0	R/(W)*	IC1MTU2CH0 High-Impedance Enable Controls the high-impedance state of the high-current pins for MTU2 CH0 when the POE0F bit is set. 0: Disables the pins to be placed in the high-impedance state. 1: Enables the pins to be placed in the high-impedance state.
0	IC1MTU2SZE	0	R/(W)*	IC1MTU2S High-Impedance Enable Controls the high-impedance state of the high-current pins for MTU2S when the POE0F bit is set. 0: Disables the pins to be placed in the high-impedance state. 1: Enables the pins to be placed in the high-impedance state.

Note: * Can be modified only once after a power-on reset.

13.4 Operation

Tables 13.4 and 13.5 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 13.4 Selection of Target Pins for High-Impedance Control with $\overline{\text{POE}}$ Input

Selection of Target Pins for High-Impedance Control	Selecting Conditions	Detailed Conditions
High-current pin for MTU2 (MTU2_hiz_1)	Selection of $\overline{\text{POE}}$ pin high-impedance control extension by default or POECR3 setting.	POE0F+IC2MTU2ZE • (POE4F) + IC3MTU2ZE • (POE8F • POE8E)
High-current pin for MTU2S (MTU2s_hiz_1)	Selection of $\overline{\text{POE}}$ pin high-impedance control extension by default or POECR3 setting.	POE4F + IC1MTU2SZE • (POE0F) + IC3MTU2SZE • (POE8F • POE8E)
High-current pin for MTU2CH0 (MTU2ch0_hiz_1)	Selection of $\overline{\text{POE}}$ pin high-impedance control extension by default or POECR3 setting.	(POE8F • POE8E) + IC1MTU2CH0ZE • (POE0F) + IC2MTU2CH0ZE • (POE4F)

Table 13.5 Target Pins and Conditions for High-Impedance Control

Pins	Conditions	Detailed Conditions
MTU2 high-current pins (PE9/TIOC3B and PE11/TIOC3D)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2P1CZE • ((MTU2_hiz_1) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PE12/TIOC4A and PE14/TIOC4C)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2P2CZE • ((MTU2_hiz_1) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PE13/TIOC4B and PE15/TIOC4D)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2P3CZE • ((MTU2_hiz_1) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2S high-current pins (PE5/TIOC3BS and PE6/TIOC3DS)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2SP1CZE • ((MTU2s_hiz_1) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PE0/TIOC4AS and PE2/TIOC4CS)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2SP2CZE • ((MTU2s_hiz_1) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PE1/TIOC4BS and PE3/TIOC4DS)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2SP3CZE • (MTU2s_hiz_1) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD10/TIOC3BS and PD11/TIOC3DS)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2SP4CZE • (MTU2s_hiz_1) +(OSF2 • OCE2) + (MTU2SHIZ))

Pins	Conditions	Detailed Conditions
MTU2S high-current pins (PD12/TIOC4AS and PD14/TIOC4CS)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2SP5CZE • (MTU2s_hiz_1) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PD13/TIOC4BS and PD15/TIOC4DS)	Input level detection of the selected $\overline{\text{POE}}$ pin, output level comparison, or SPOER setting	MTU2SP6CZE • (MTU2s_hiz_1) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2 CH0 pins (PE0/TIOC0A, PE1/TIOC0B, PE2/TIOC0C, and PE3/TIOC0D)	Input level detection of the selected $\overline{\text{POE}}$ pin detection or SPOER setting	MTU2PE0ZE to MTU2PE3ZE • (MTU2ch0_hiz_1) +(MTU2CH0HIZ)
MTU2 CH0 pins (PB1/TIOC0A, PB2/TIOC0B, PB3/TIOC0C, and PB4/TIOC0D)	Input level detection of the selected $\overline{\text{POE}}$ pin or SPOER setting	MTU2PB1ZE to MTU2PB4ZE • (MTU2ch0_hiz_1) +(MTU2CH0HIZ)

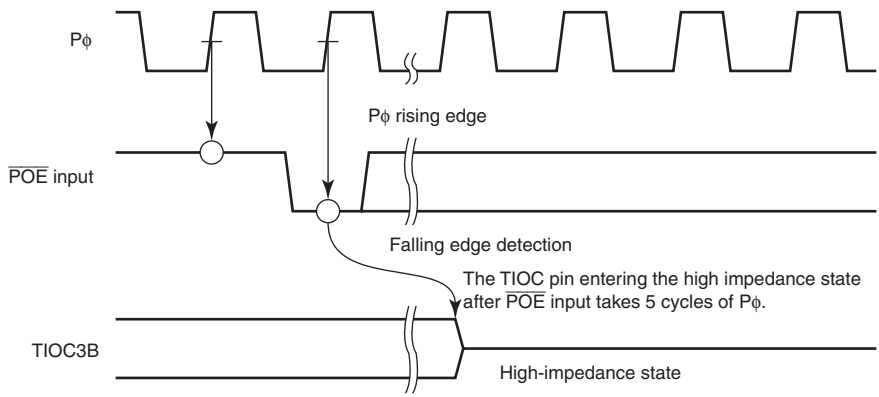
13.4.1 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR3 occur on the $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ 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 $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ 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 $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ 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.

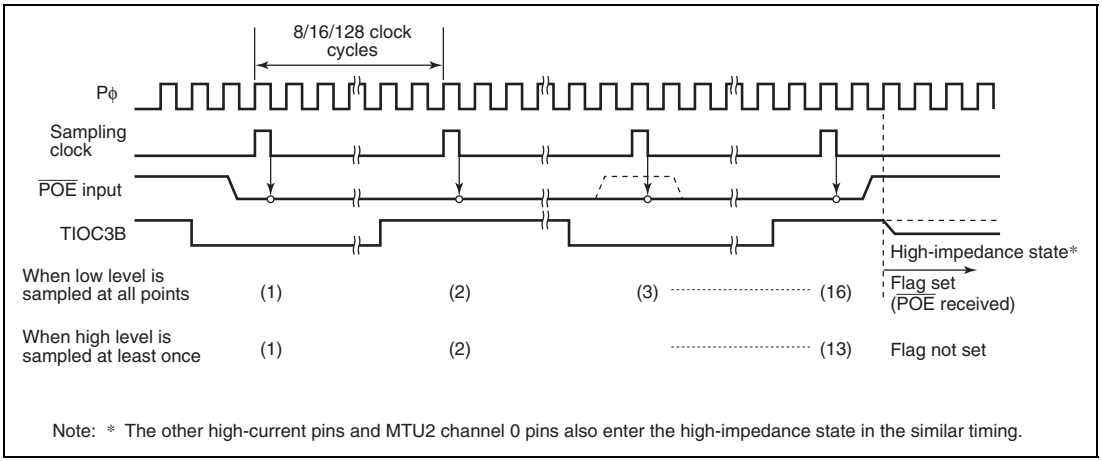


Figure 13.3 Low-Level Detection Operation

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.

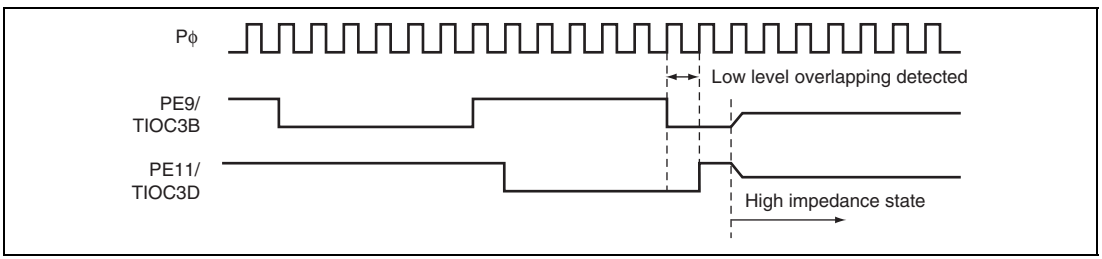


Figure 13.4 Output-Level Compare Operation

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 the flag in bit 12 (POE8F, POE4F, and POE0F) of ICSR1 to ICSR3. However, note that when low-level sampling is selected by bits 1 and 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 $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ 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 13.6 shows the interrupt sources and their conditions.

Table 13.6 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F and OSF1	PIE1 • 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.

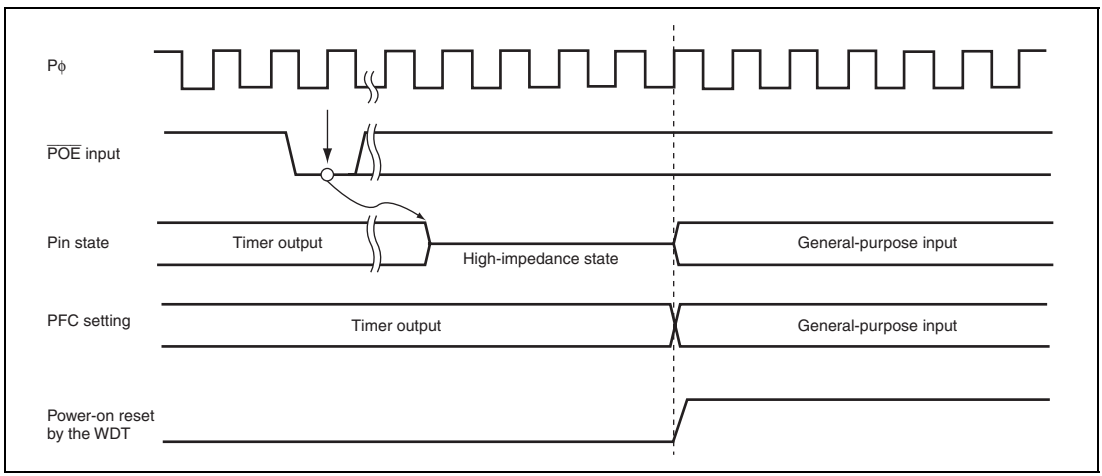


Figure 13.5 Pin States when the Watchdog Timer Issues a Power-on Reset

13.6.2 Input Pins

When the $\overline{\text{POE}}$ function is to be used, input a logical 1 to the $\overline{\text{POE0}}$, $\overline{\text{POE4}}$ and $\overline{\text{POE8}}$ 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.

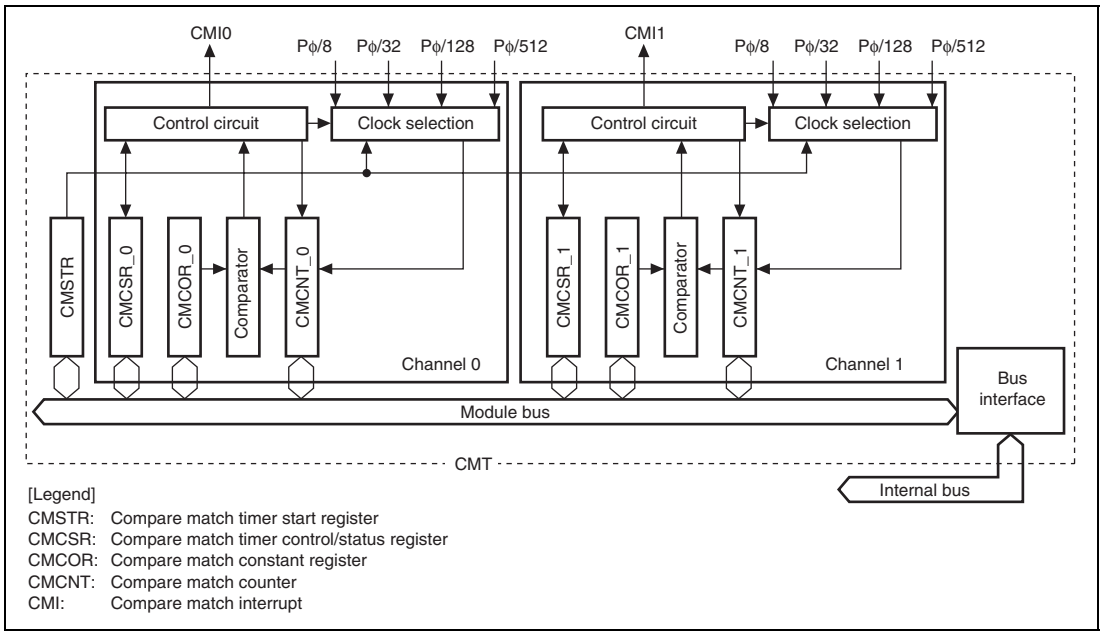


Figure 14.1 Block Diagram of CMT

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. 1: CMCNT and CMCOR values 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
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.

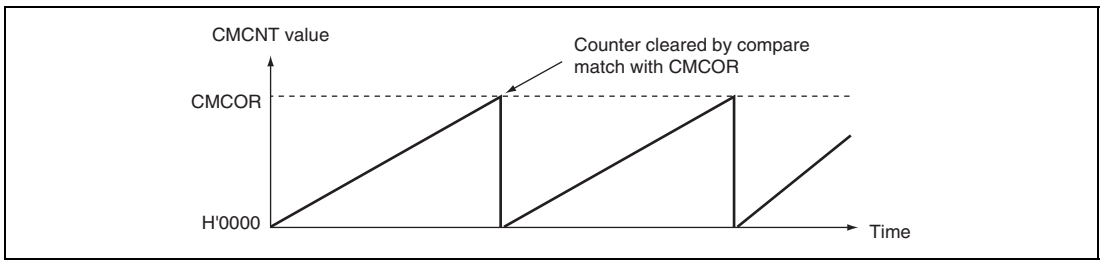


Figure 14.2 Counter Operation

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.

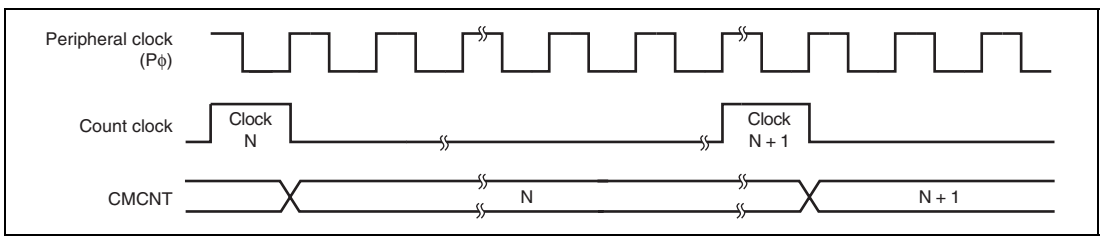


Figure 14.3 Count Timing

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 Bit	DMAC/DTC Activation	Priority Order
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.

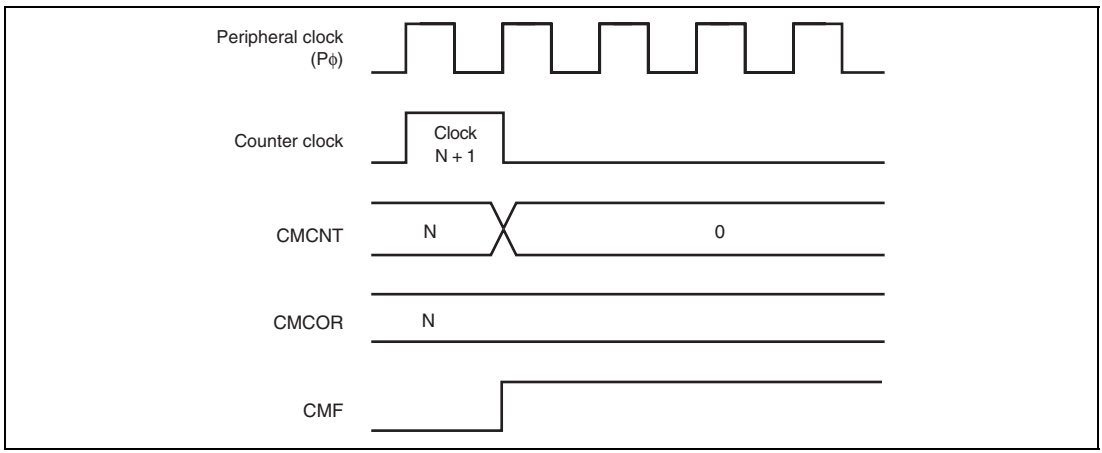


Figure 14.4 Timing of CMF Setting

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. Furthermore, exit from the interrupt handler should follow checking to ensure that the CMF bit has been cleared.

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.

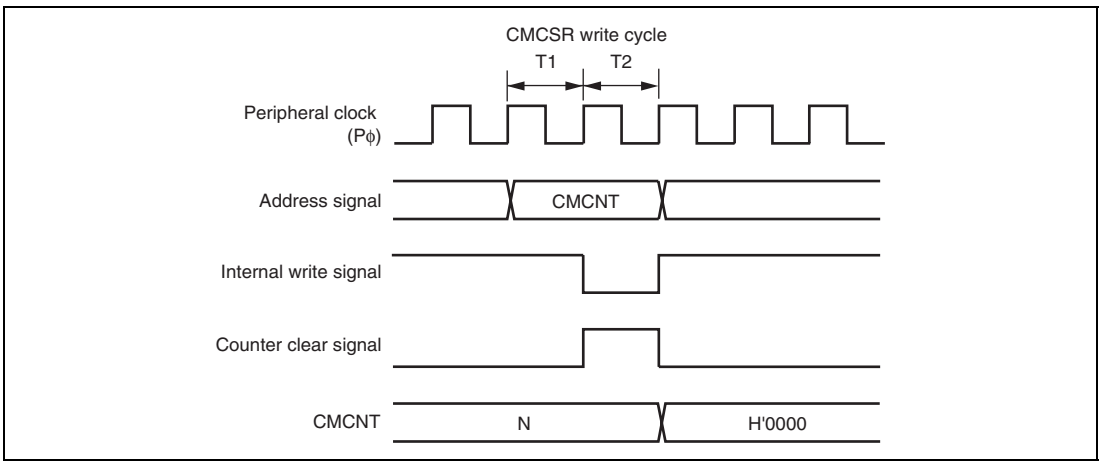


Figure 14.5 Conflict between Write and Compare Match Processes of CMCNT

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.

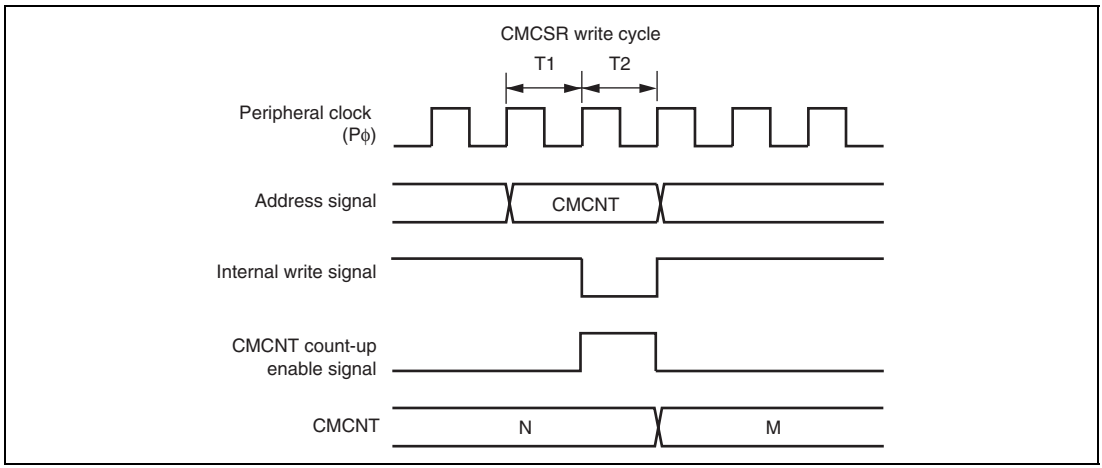


Figure 14.6 Conflict between Word-Write and Count-Up Processes of CMCNT

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.

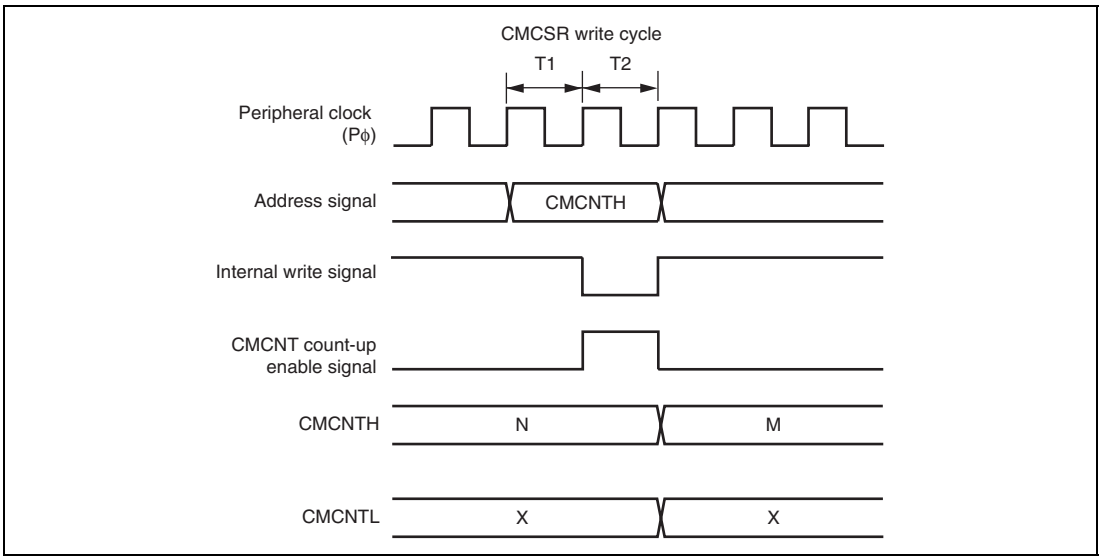


Figure 14.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

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.

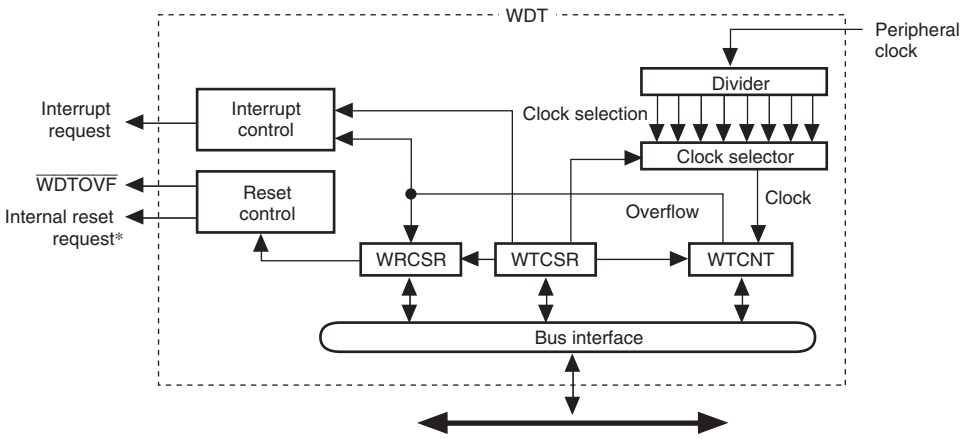


Figure 15.1 Block Diagram of WDT

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 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. 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.

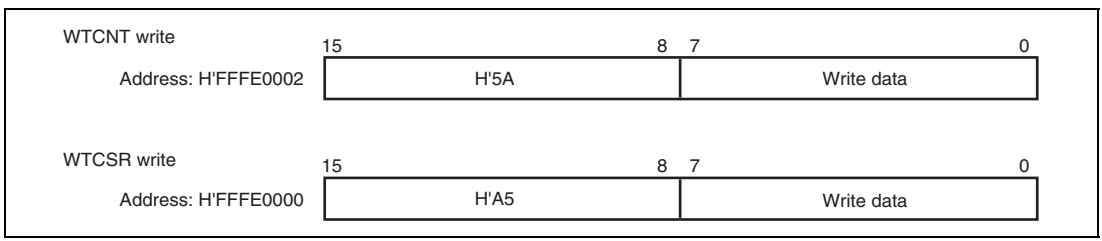


Figure 15.2 Writing to WTCNT and WTCSR

(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.

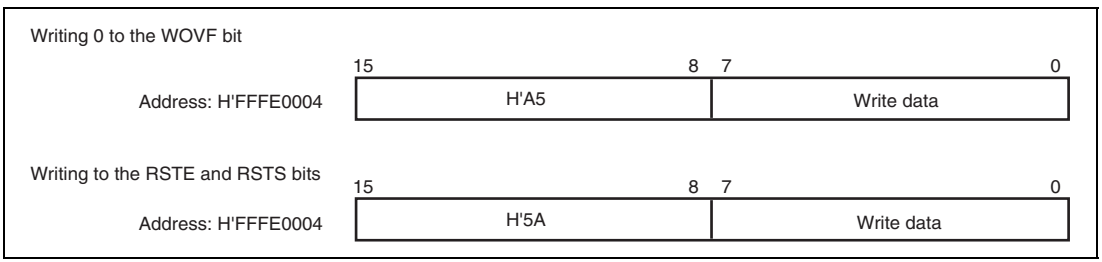


Figure 15.3 Writing to WRCSR

(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 26, 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 $64 \times \text{P}\phi$ 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 $128 \times \text{P}\phi$ 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.

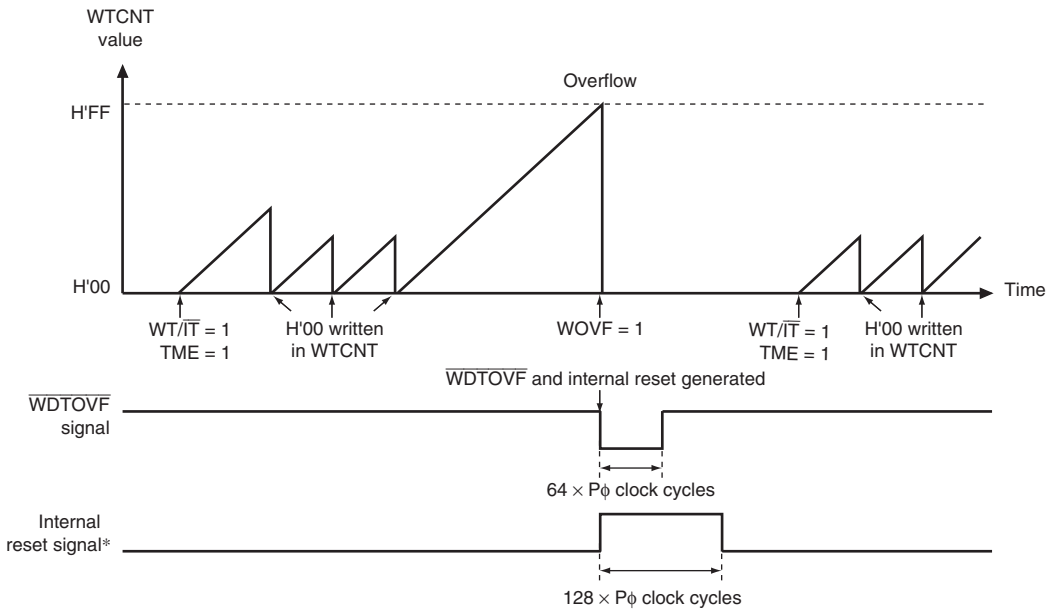


Figure 15.4 Operation in Watchdog Timer Mode

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.

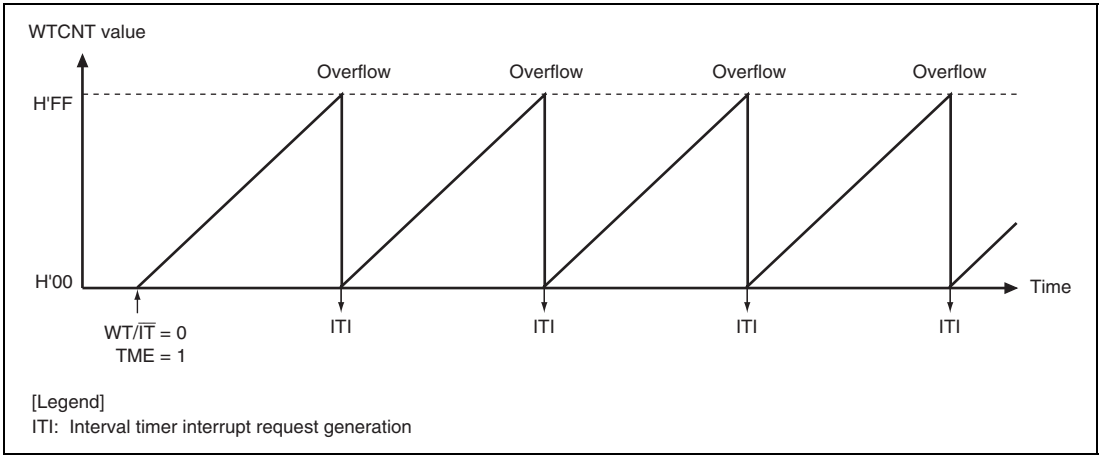


Figure 15.5 Operation in Interval Timer Mode

15.5 Interrupt Source

The WDT issues an interval timer interrupt (ITI).

Table 15.3 shows the interrupt source. The interval timer interrupt is generated when the interval timer overflow flag bit (IOVF) in the watchdog timer control status register (WTCSR) is set to 1.

The interrupt request is released by clearing the interrupt flag bit to 0.

Table 15.3 Interrupt Source

Name	Interrupt Source	Interrupt Flag Bit
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 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.

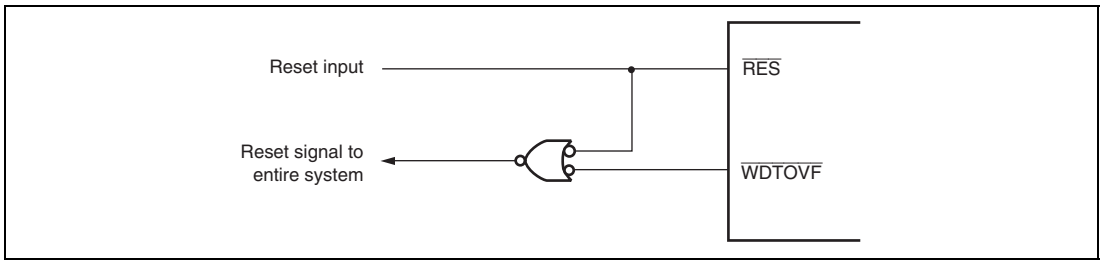


Figure 15.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

15.6.4 Manual Reset in Watchdog Timer Mode

When a manual reset occurs in watchdog timer mode, the 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.

15.6.5 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)

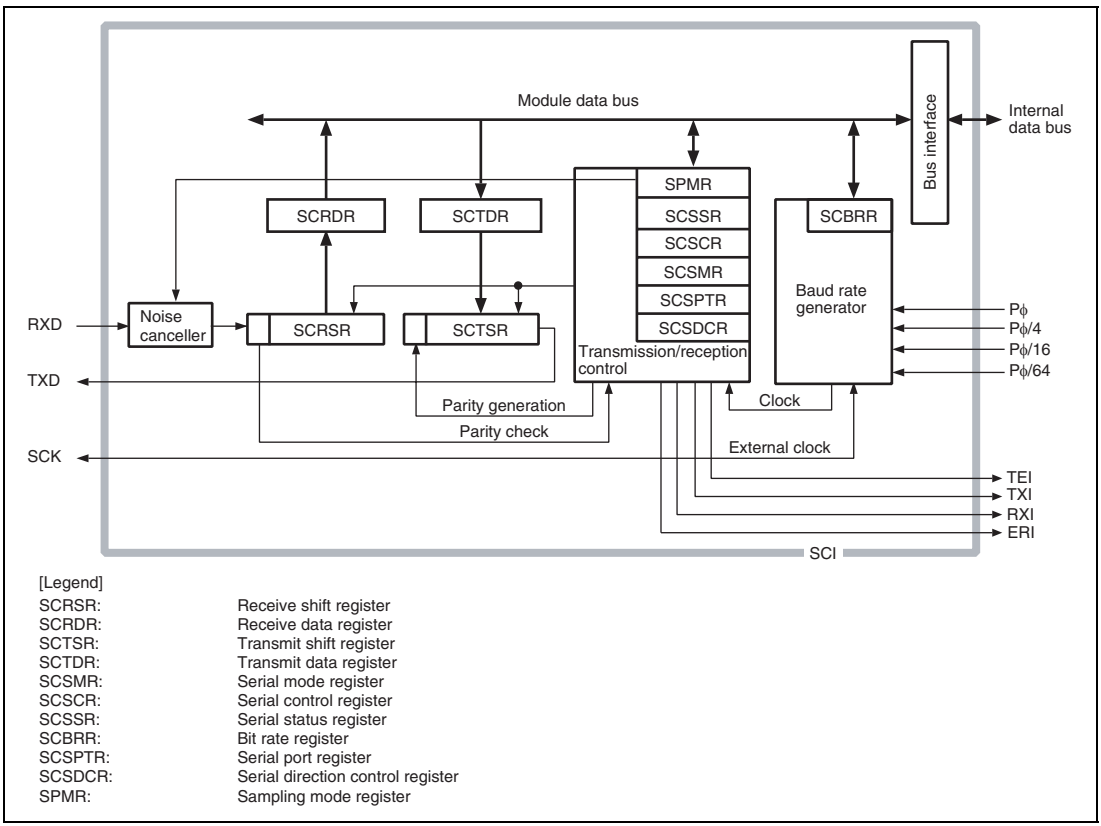
This LSI has three channels of independent serial communication interface (SCI). The SCI can handle both asynchronous and clock synchronous serial communication. In asynchronous serial communication mode, serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication 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 SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication 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 SCI 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.
- Noise canceller (for asynchronous communication only)

Figure 16.1 shows a block diagram of the SCI.



[Legend]

SCRSR:	Receive shift register
SCRDR:	Receive data register
SCTSR:	Transmit shift register
SCTDR:	Transmit data register
SCSMR:	Serial mode register
SCSCR:	Serial control register
SCSSR:	Serial status register
SCBRR:	Bit rate register
SCSPTR:	Serial port register
SCSDCR:	Serial direction control register
SPMR:	Sampling mode register

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

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 28, 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
	Sampling monitor register_0	SPMR_0	R/W	H'00	H'FFFF8014	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
	Sampling monitor register_1	SPMR_1	R/W	H'00	H'FFFF8814	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
	Sapling monitor register_2	SPMR_2	R/W	H'00	H'FFFF9014	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 TEND after reading TEND = 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 22, 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:

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

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)

	Pϕ (MHz)																	
Bit Rate	10*1			12*1			14*1			16*1			18*1			20		
(bits/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
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*[1]			30*[1]			32*[1]		
	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*1			36*1			38*1			40			50*2		
	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
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

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

Table 16.7 Bit Rates and SCBRR Settings in Clock Synchronous Mode (1)

Bit Rate (bits/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
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*2	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*2

Table 16.8 Bit Rates and SCBRR Settings in Clock Synchronous Mode (2)

Bit Rate (bits/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
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*1		36*1		38*1		40		50*3	
	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		

Notes: Settings with an error of 1% or less are recommended.

1. Cannot be set for this LSI.
2. Continuous transmission/reception is disabled.
3. This is only available for the SH7239B and SH7237B.

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

Table 16.10 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 16.11 and 16.12 list the maximum rates for external clock input.

Table 16.10 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
10*[1]	312500	0	0
12*[1]	375000	0	0
14*[1]	437500	0	0
16*[1]	500000	0	0
18*[1]	562500	0	0
20	625000	0	0
22	687500	0	0
24	750000	0	0
26*[1]	812500	0	0
28*[1]	875000	0	0
30*[1]	937500	0	0
32*[1]	1000000	0	0
34*[1]	1062500	0	0
36*[1]	1125000	0	0
38*[1]	1187500	0	0
40	1250000	0	0
50*[2]	1562500	0	0

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

Table 16.11 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10*[1]	2.5000	156250
12*[1]	3.0000	187500
14*[1]	3.5000	218750
16*[1]	4.0000	250000
18*[1]	4.5000	281250
20	5.0000	312500
22	5.5000	343750
24	6.0000	375000
26*[1]	6.5000	406250
28*[1]	7.0000	437500
30*[1]	7.5000	468750
32*[1]	8.0000	500000
34*[1]	8.5000	531250
36*[1]	9.0000	562500
38*[1]	9.5000	593750
40	10.0000	625000
50*[2]	12.5000	781250

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

Table 16.12 Maximum Bit Rates with External Clock Input (Clock Synchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
10*[1]	1.6667	1666666.7
12*[1]	2.0000	2000000.0
14*[1]	2.3333	2333333.3
16*[1]	2.6667	2666666.7
18*[1]	3.0000	3000000.0
20	3.3333	3333333.3
22	3.6667	3666666.7
24	4.0000	4000000.0
26*[1]	4.3333	4333333.3
28*[1]	4.6667	4666666.7
30*[1]	5.0000	5000000.0
32*[1]	5.3333	5333333.3
34*[1]	5.6667	5666666.7
36*[1]	6.0000	6000000.0
38*[1]	6.3333	6333333.3
40	6.6667	6666666.7
50*[2]	8.3333	8333333.3

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

16.3.11 Sampling Mode Register (SPMR)

SPMR enables or disables the noise canceller function during asynchronous communication.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	STD SPM
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 1	R	Reserved These bits are always read as 1. The write value should always be 1.
0	STDSPM	0	R/W	Noise Cancellation Function Select Selects the noise cancellation function for the RXD pin input when asynchronous communication is performed. 0: Noise canceller is disabled. 1: Noise canceller is enabled. This bit should be written to when RE is 0.

- Noise Canceller

  The RXD input signal is loaded internally via the noise canceller. The noise canceller circuit consists of three-stage F/F circuits connected in series and a match-detection circuit. The RXD input signal is sampled on a basic clock with a frequency 16-times the transfer rate, and the level is passed forward to the next circuit when outputs of three F/Fs match. When the outputs do not match, previous value is retained.

  In other word, when the same level is retained for three clock cycles or more, the input signal is acknowledged as a signal. When the level is changed within three clock cycles, the change is acknowledged as not a signal change but noise.

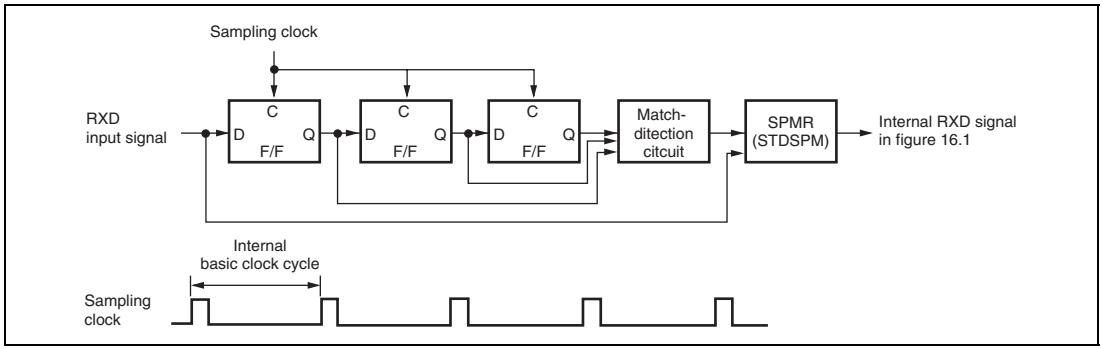


Figure 16.2 Block Diagram of Noise Canceller

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 16.13. 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 16.14.

(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 16.13 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 16.14 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 16.3 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.

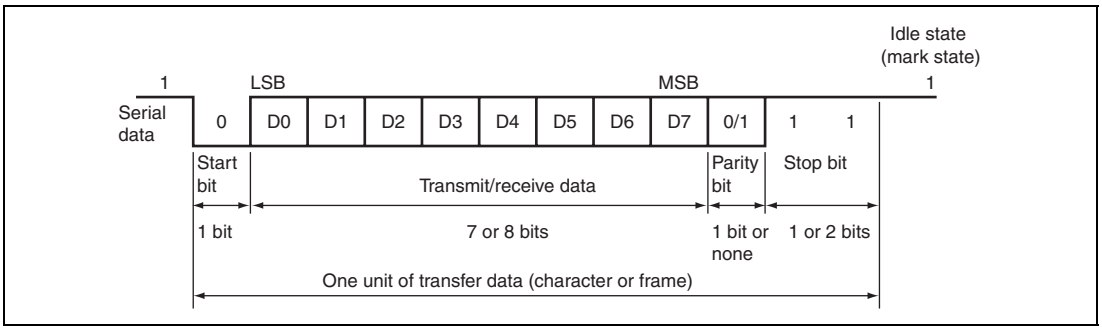


Figure 16.3 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 16.15 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 16.15 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 16.14).

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.

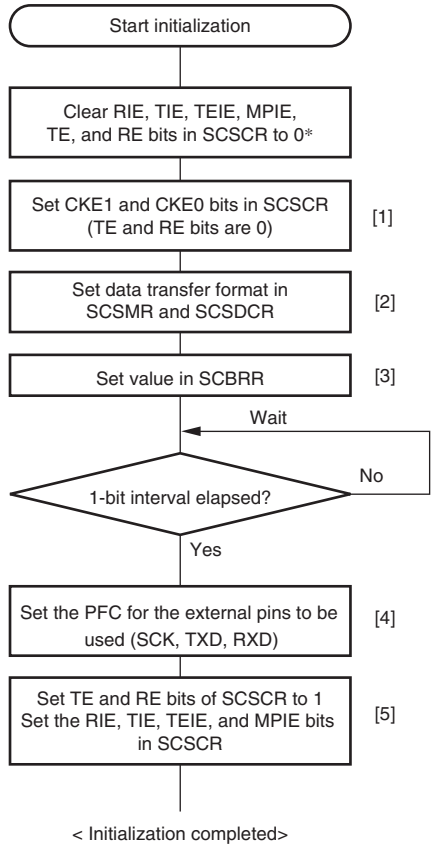
(3) Transmitting and Receiving Data

- 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. Ensure that settings for the PFC are only made after the settings in steps 1 to 3 below.

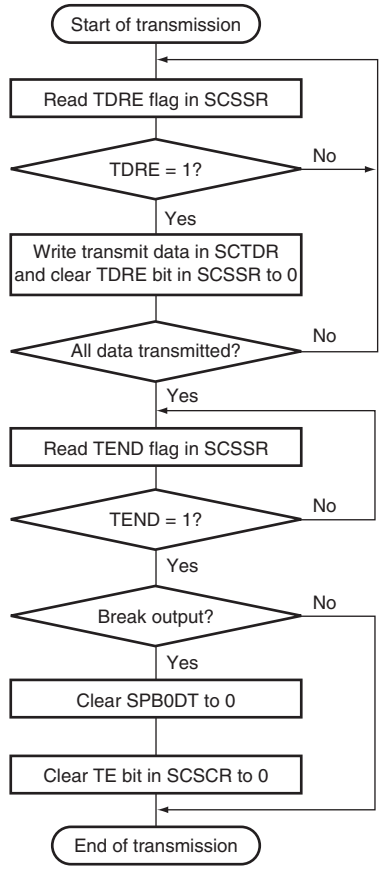
[5] Set the TE bit or RE bit in SCSCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting, and RXD pin is in an idle state for waiting the start bit during receiving.

Note : * In simultaneous transmit/receive operation, the TE and RE bits must be cleared to 0 or set to 1 simultaneously.

Figure 16.4 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Asynchronous Mode)

Figure 16.5 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 16.5 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 16.6 shows an example of the operation for transmission.

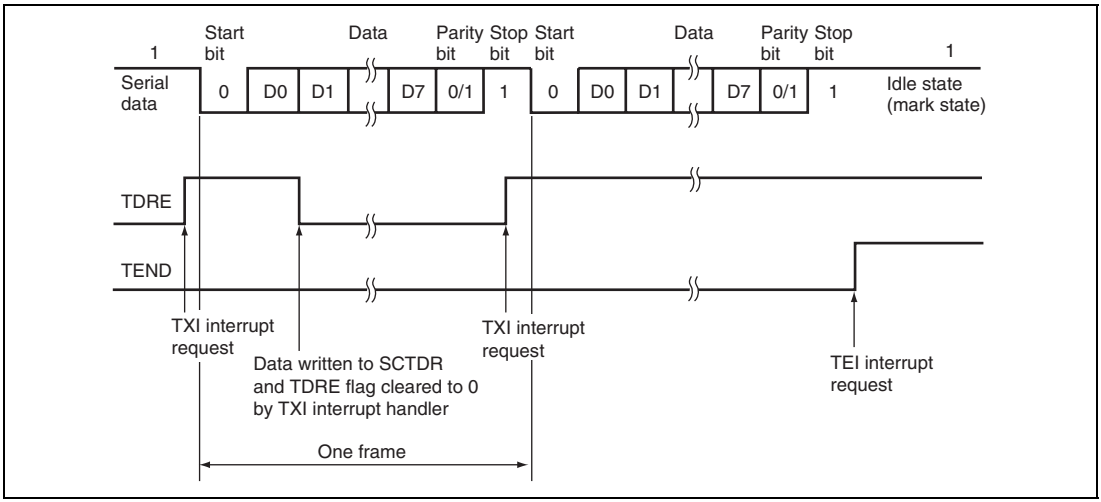


Figure 16.6 Example of Transmission in Asynchronous Mode (8-Bit Data, Parity, One Stop Bit)

- Receiving Serial Data (Asynchronous Mode)

Figure 16.7 shows a sample flowchart for serial reception. Use the following procedure for serial data reception after enabling the SCI for reception.

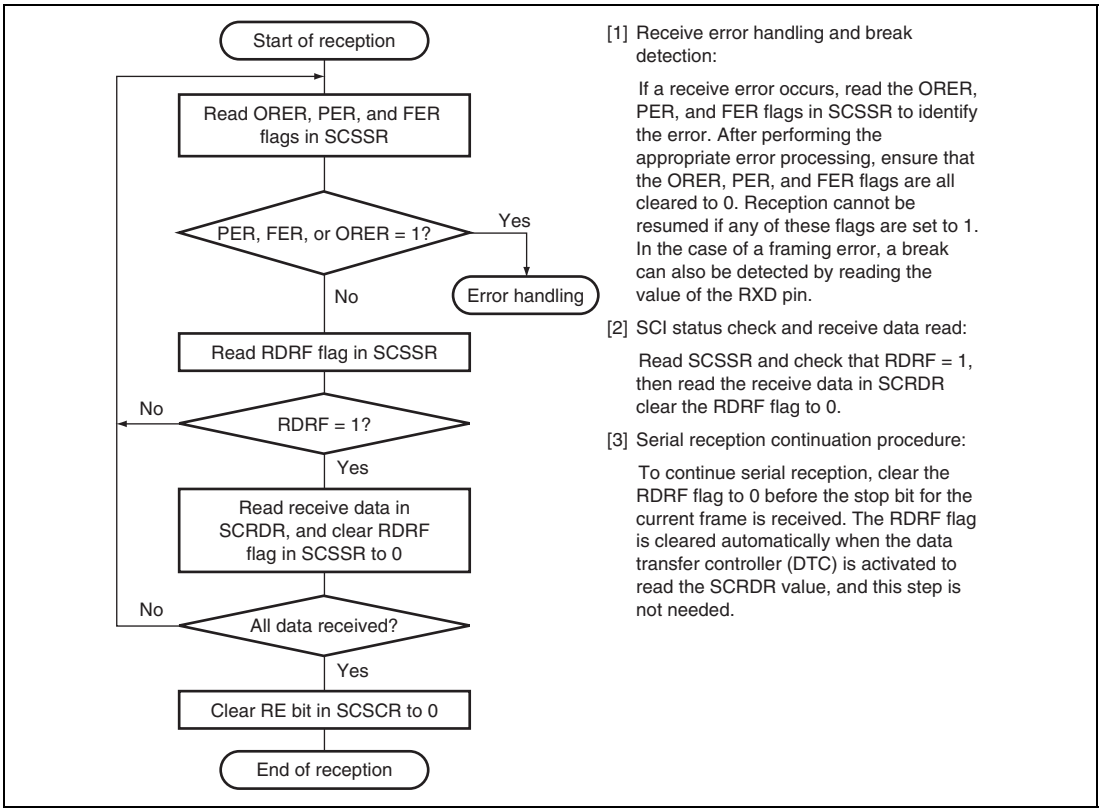


Figure 16.7 Sample Flowchart for Receiving Serial Data (1)

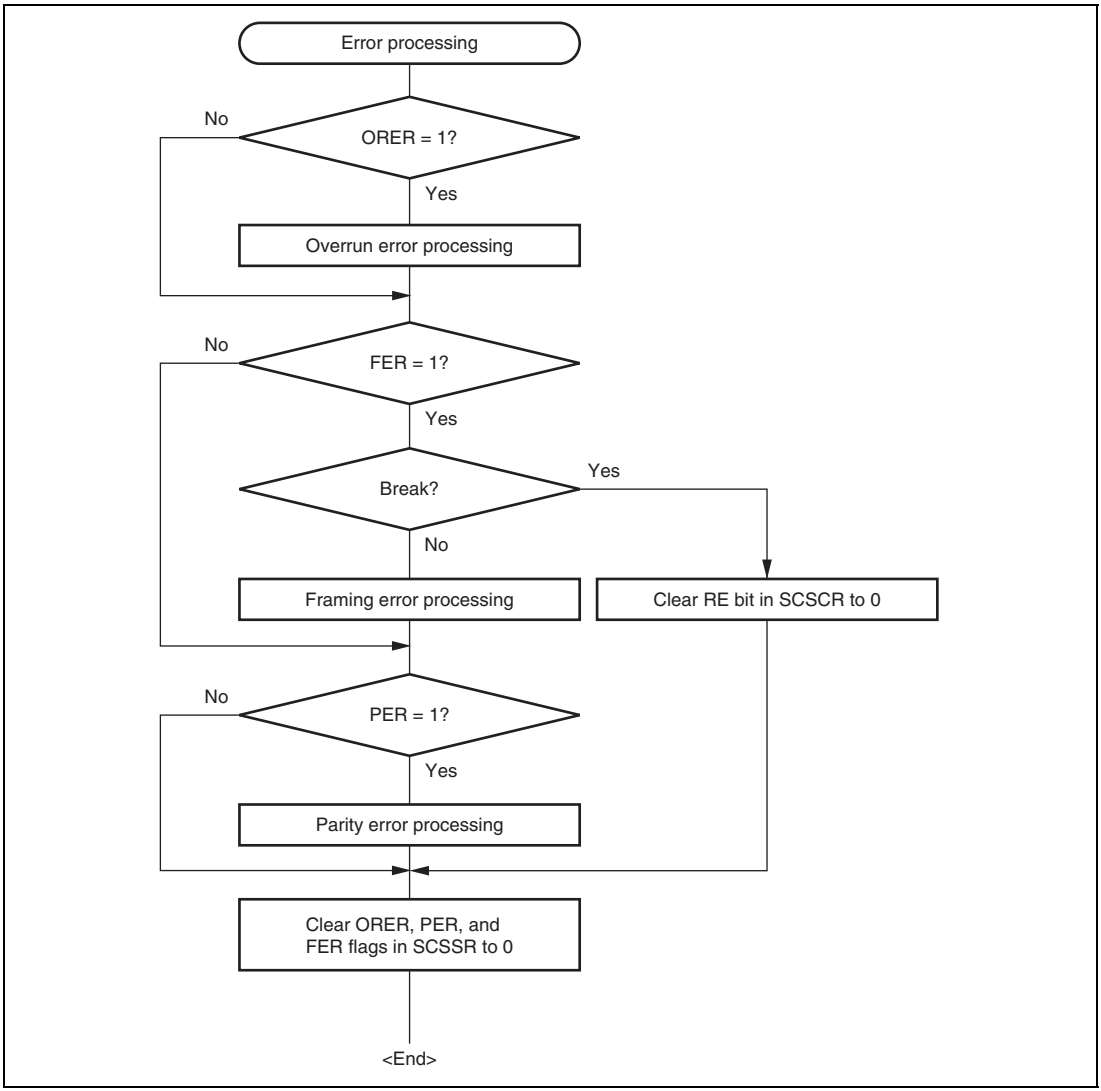


Figure 16.7 Sample Flowchart for Receiving Serial Data (2)

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 16.16.

   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 16.16 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 16.8 shows an example of the operation for reception.

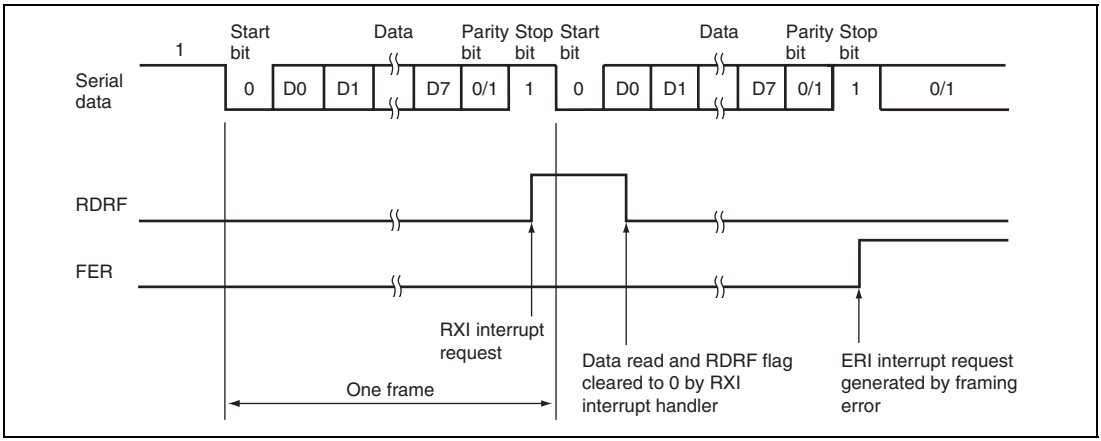


Figure 16.8 Example of SCI Receive Operation (8-Bit Data, Parity, One Stop Bit)

16.4.3 Clock Synchronous Mode

In clock synchronous mode, the SCI 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 16.9 shows the general format in clock synchronous serial communication.

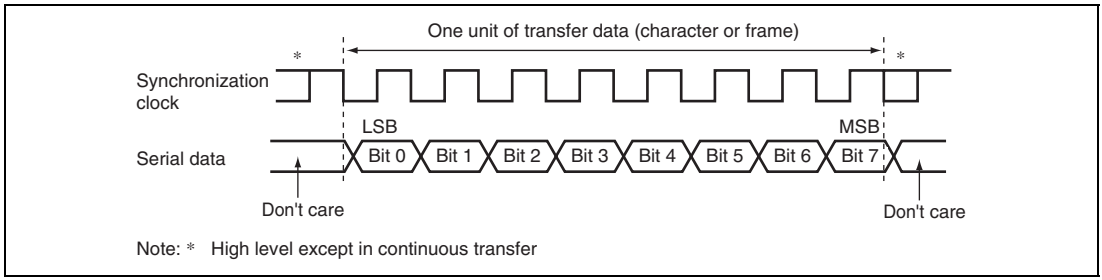


Figure 16.9 Data Format in Clock Synchronous Communication

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 16.14.

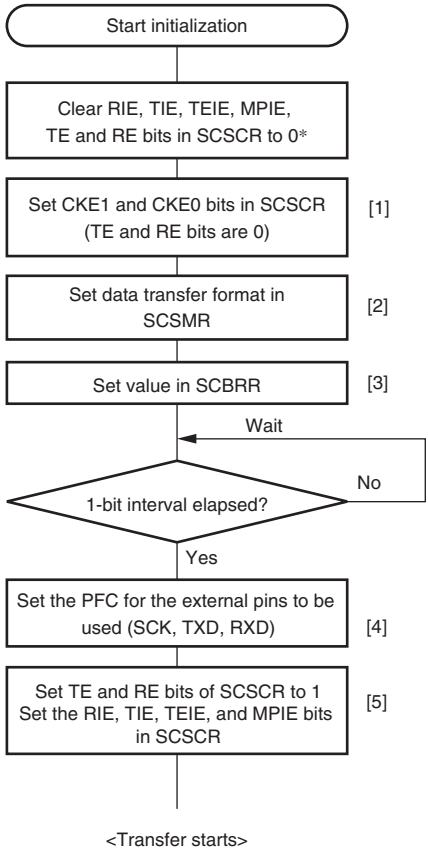
When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains in the high state. However, in reception-only operation, the synchronizing 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 16.10 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.

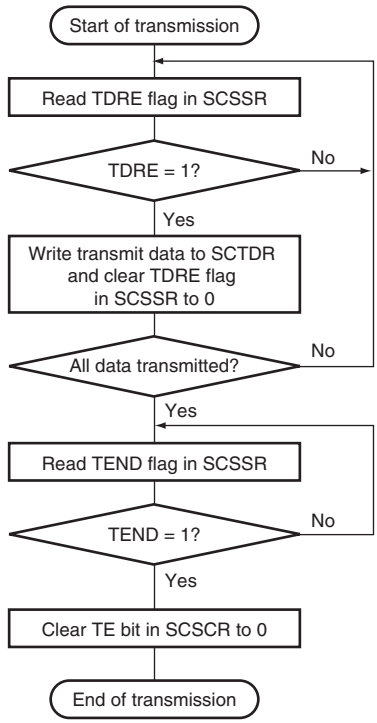
[5] Set the TE bit or RE bit in SCR to 1.* Also make settings of the RIE, TIE, TEIE, and MPIE bits. At this time, the TXD, RXD, and SCK pins are ready to be used. The TXD pin is in a mark state during transmitting. When synchronous clock output (clock master) is set during receiving in clock synchronous 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 16.10 Sample Flowchart for SCI Initialization

- Transmitting Serial Data (Clock Synchronous Mode)

Figure 16.11 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 16.11 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 16.12 shows an example of SCI transmit operation.

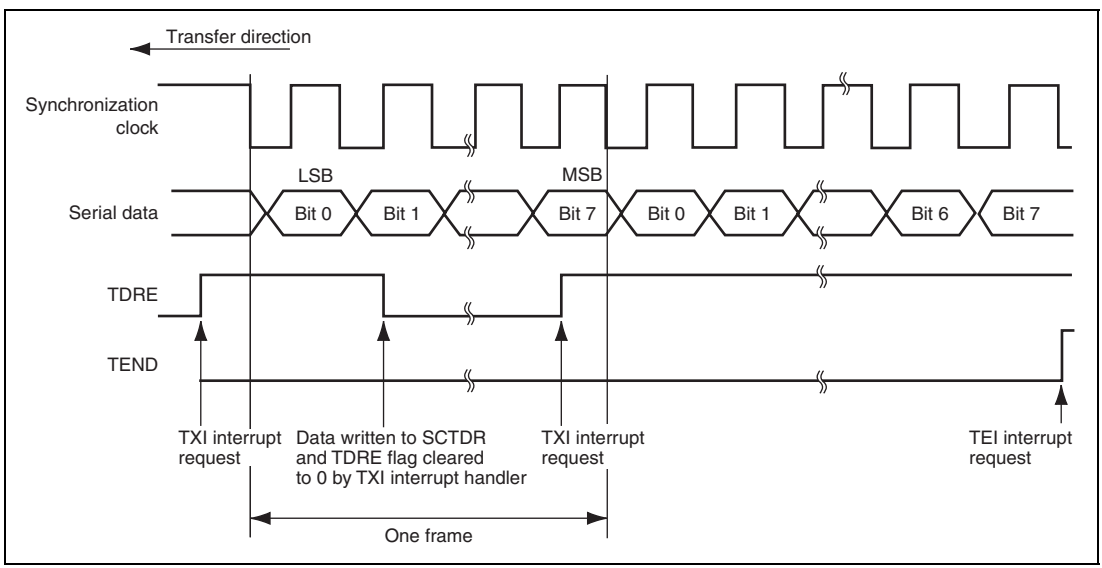
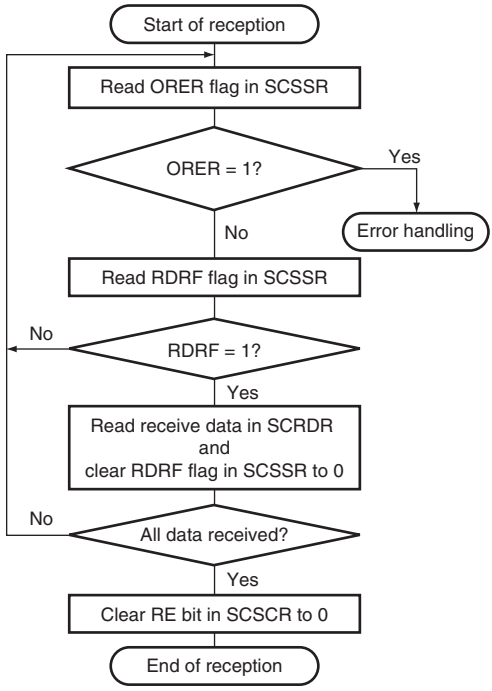


Figure 16.12 Example of SCI Transmit Operation

- Receiving Serial Data (Clock Synchronous Mode)

Figure 16.13 shows a sample flowchart for receiving serial data. Use the following procedure for serial data reception after enabling the SCI 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 16.13 Sample Flowchart for Receiving Serial Data (1)

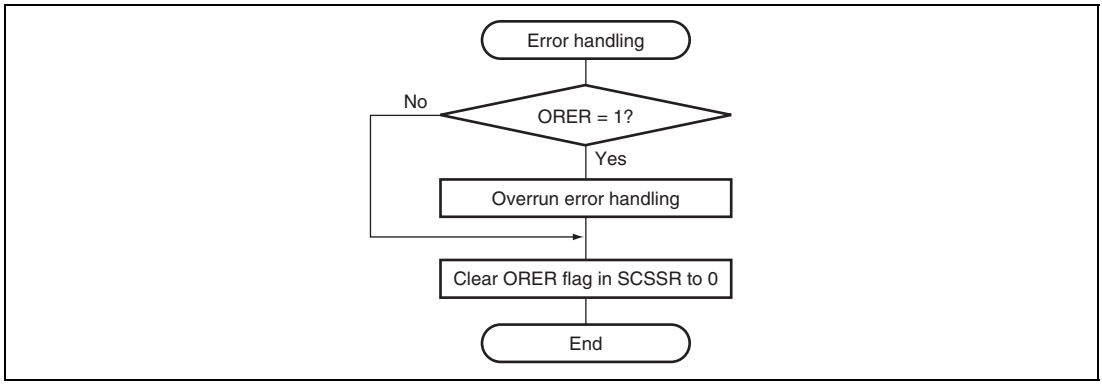


Figure 16.13 Sample Flowchart for Receiving Serial Data (2)

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 16.16. 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 16.14 shows an example of SCI receive operation.

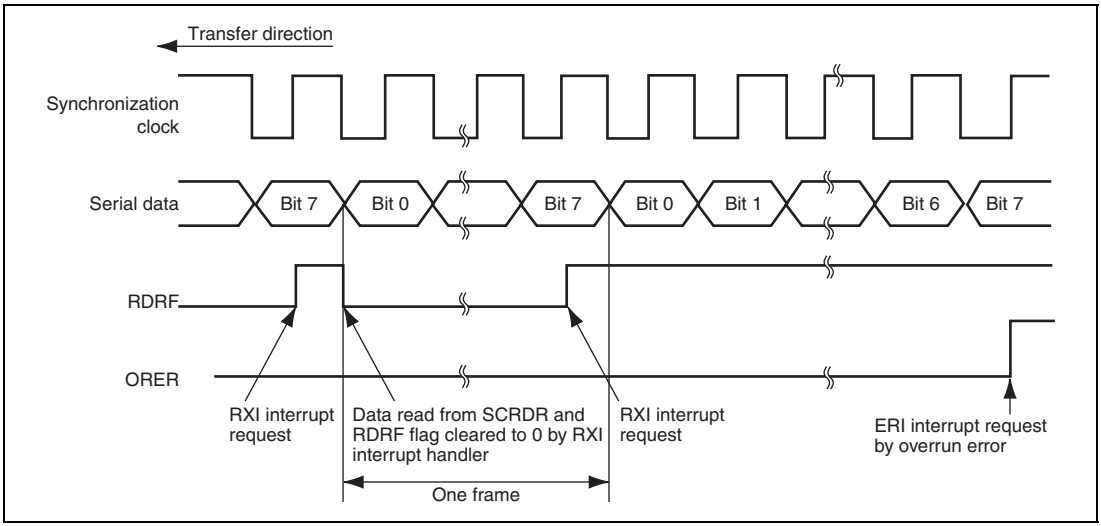
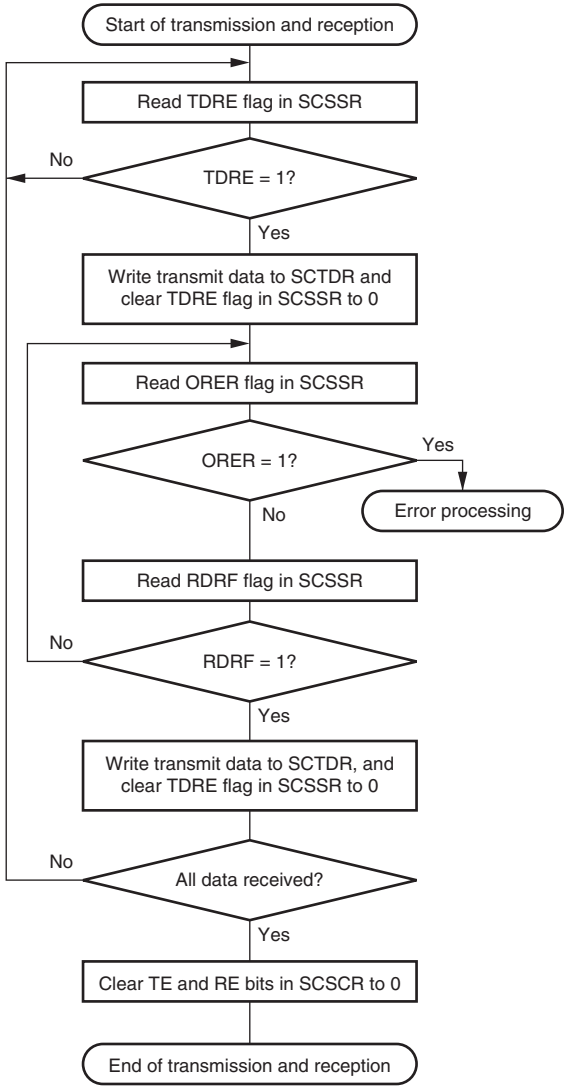


Figure 16.14 Example of SCI Receive Operation

- Transmitting and Receiving Serial Data Simultaneously (Clock Synchronous Mode)

Figure 16.15 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 16.15 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 16.16 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 MPBT 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.

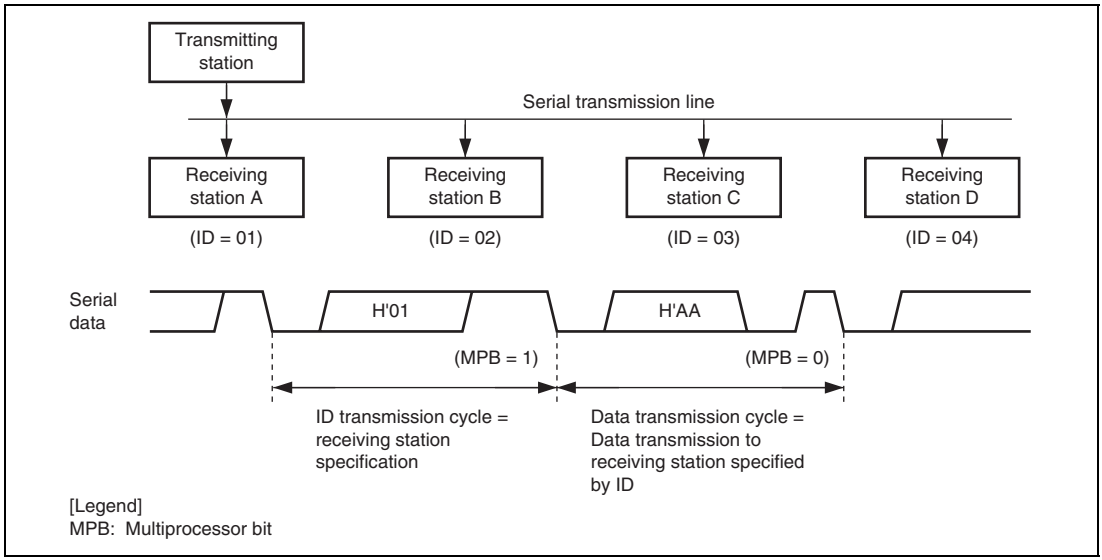
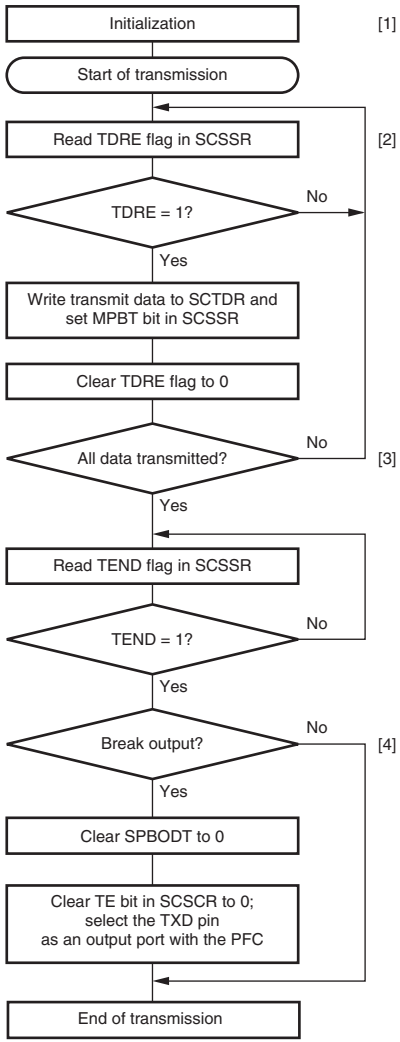


Figure 16.16 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

16.4.5 Multiprocessor Serial Data Transmission

Figure 16.17 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 16.17 Sample Multiprocessor Serial Transmission Flowchart

16.4.6 Multiprocessor Serial Data Reception

Figure 16.19 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 16.18 shows an example of SCI operation for multiprocessor format reception.

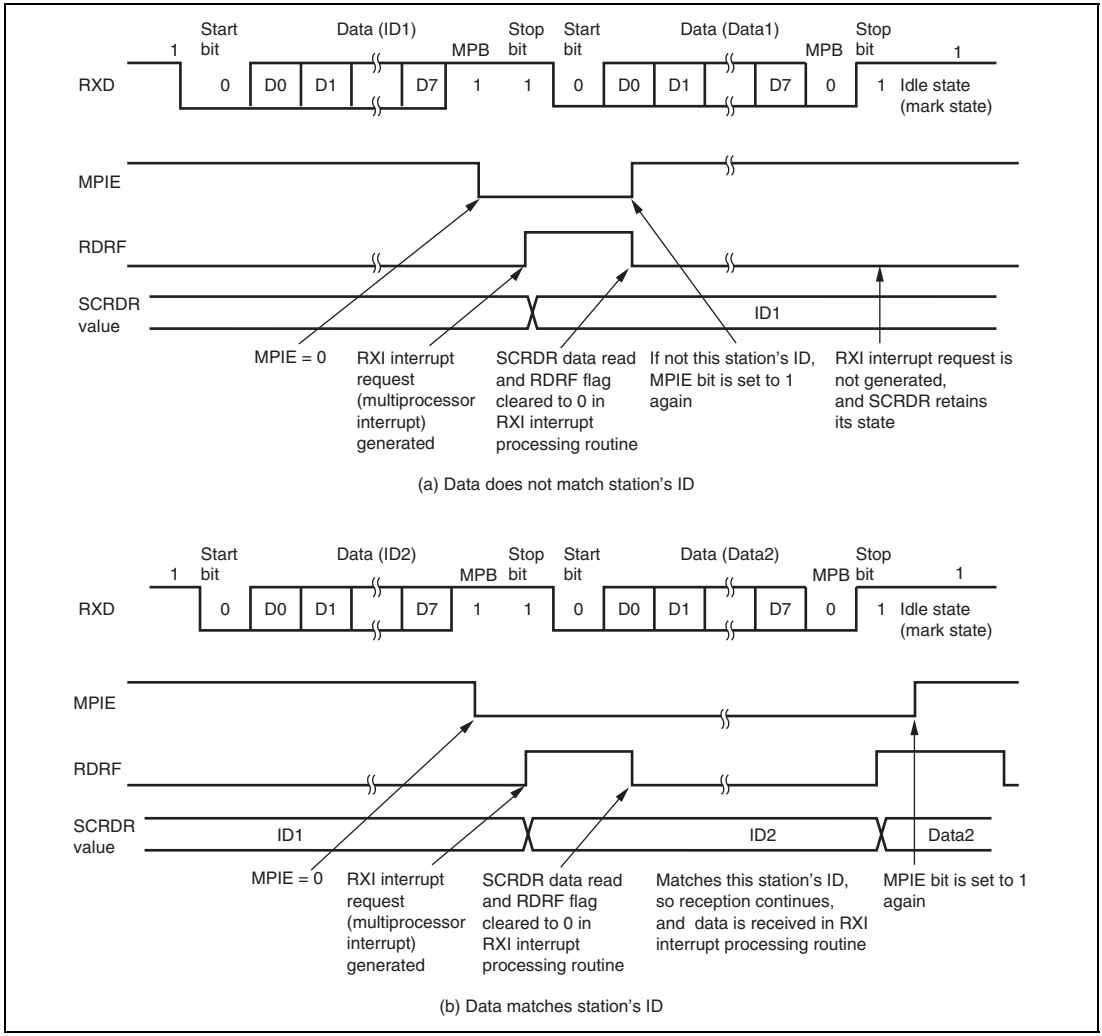
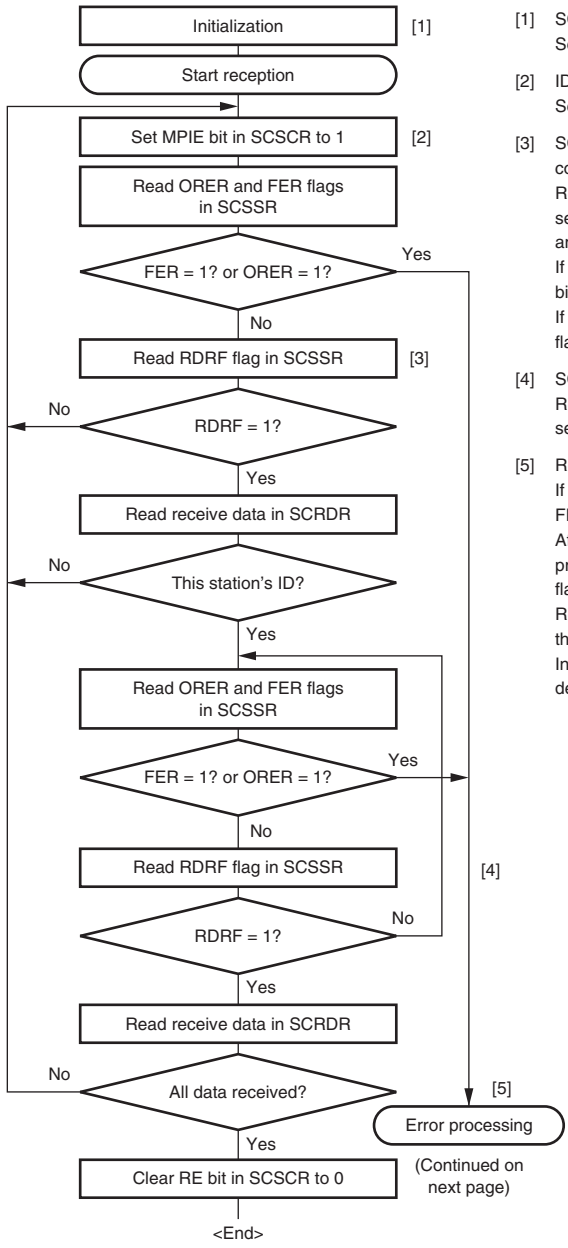


Figure 16.18 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)



[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 16.19 Sample Multiprocessor Serial Reception Flowchart (1)

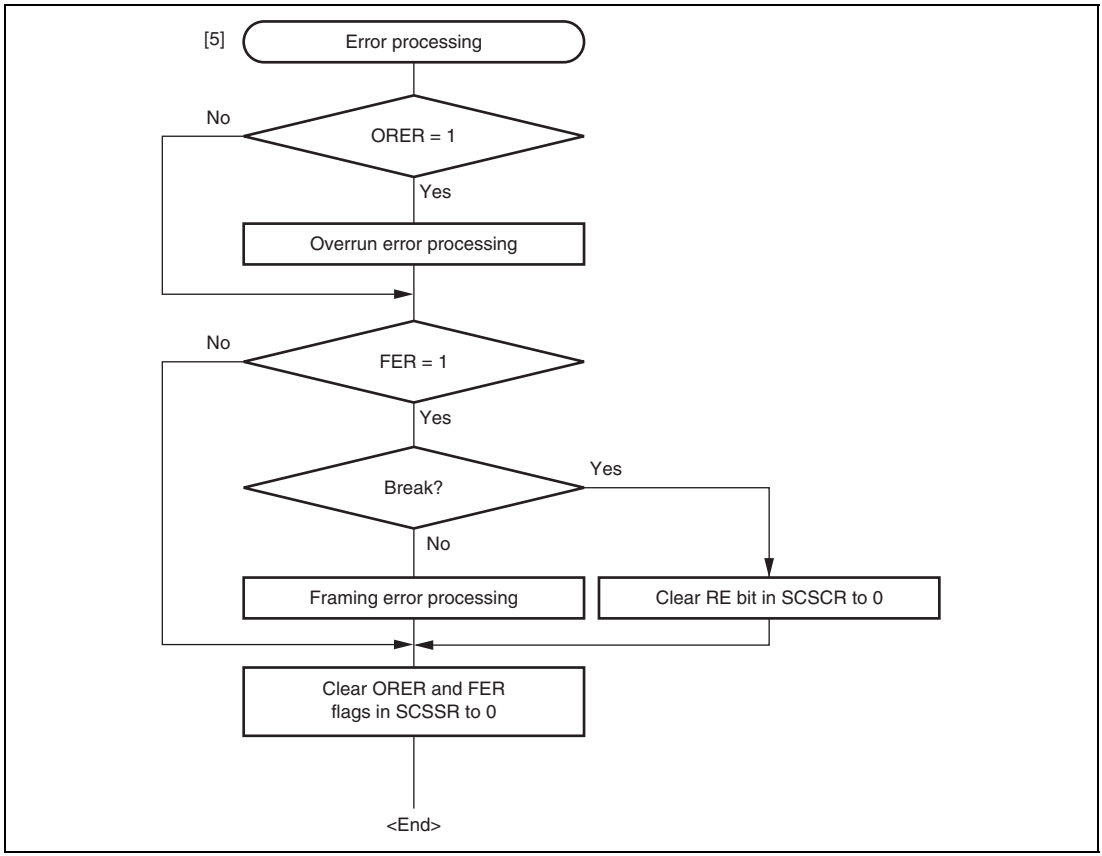


Figure 16.19 Sample Multiprocessor Serial Reception Flowchart (2)

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 16.17 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 16.17 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 16.20 and 16.21.

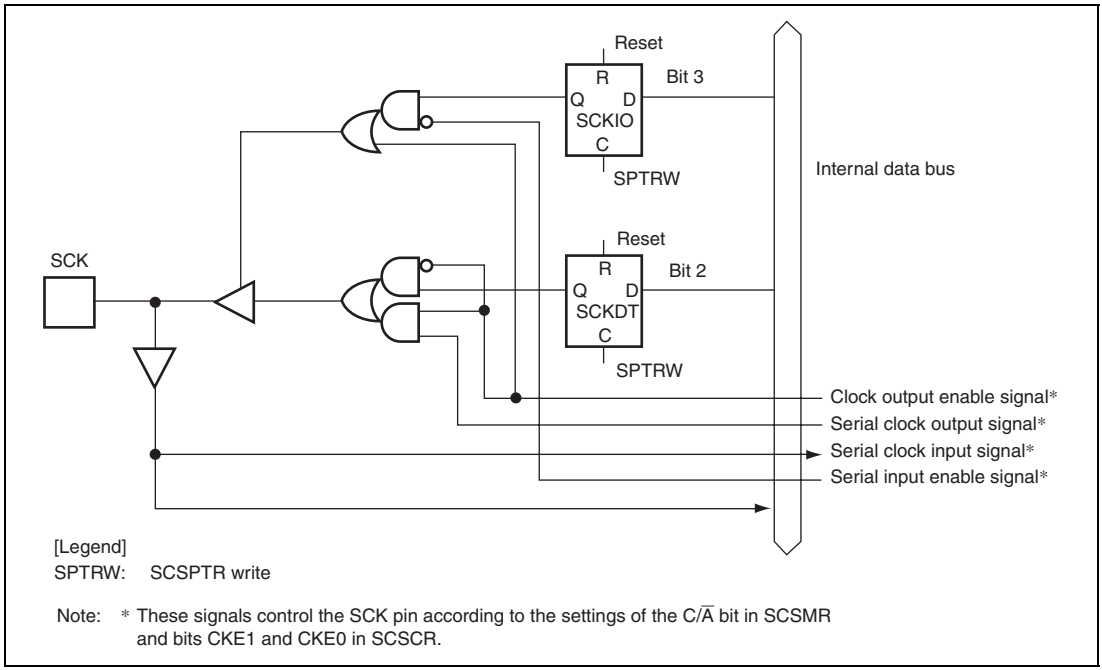


Figure 16.20 SCKIO Bit, SCKDT Bit, and SCK Pin

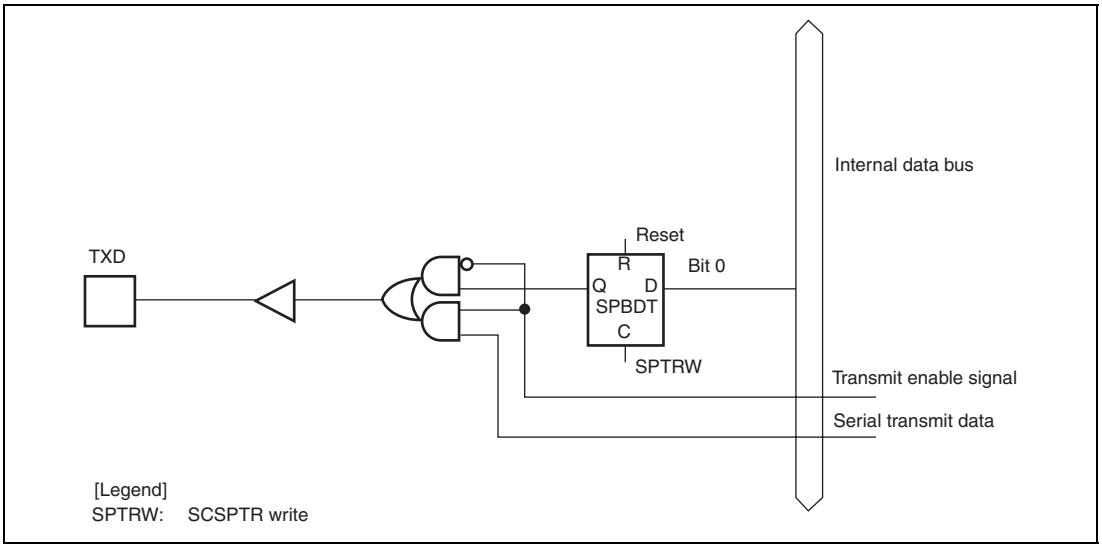


Figure 16.21 SPBDT Bit and TXD Pin

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 16.18. 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 16.18 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 16.22.

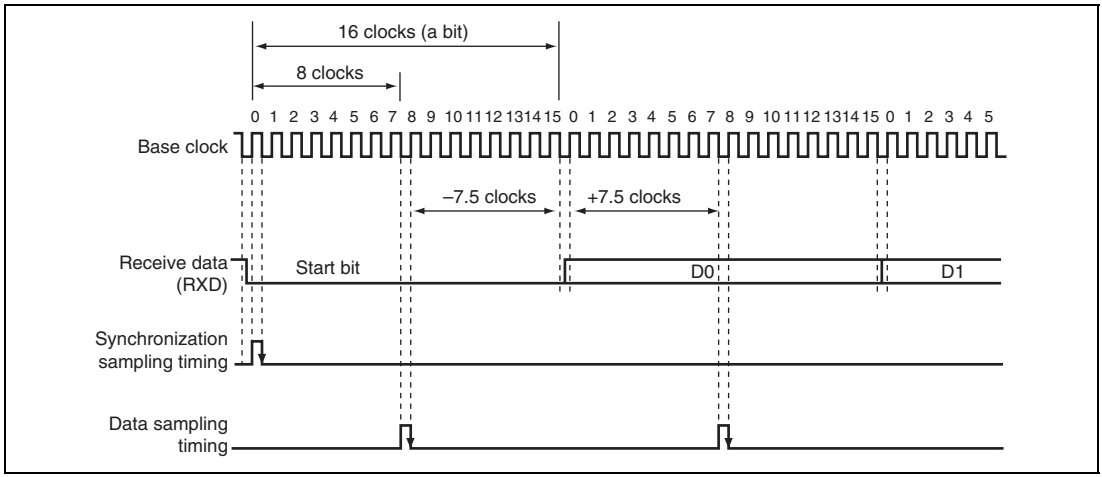


Figure 16.22 Receive Data Sampling Timing in Asynchronous Mode

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 16.23).

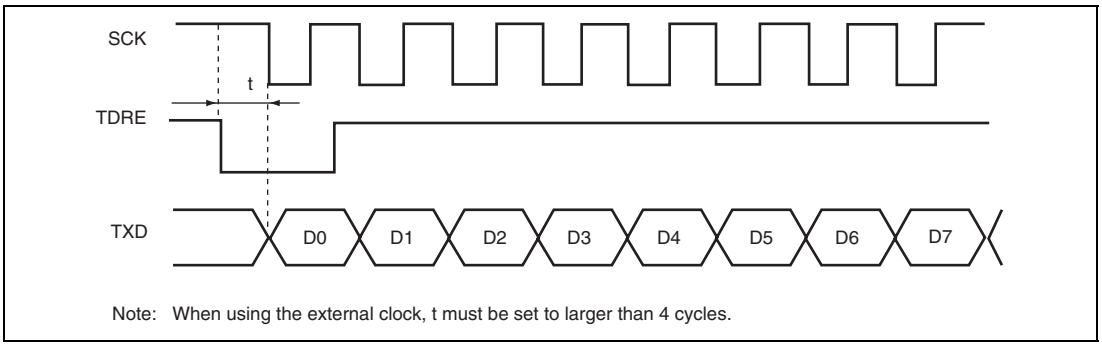


Figure 16.23 Example of Clock Synchronous Transfer Using DTC

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 26, Power-Down Modes.

16.7.9 Note for RXD Pin State on Setting RE Bit

To use the SCI, be sure to drive the RXD pin state high before setting the RE bit to 1. If the RE bit is set to 1 with the RXD pin being low, reception may be started.

16.7.10 Clearing Interrupt Flags

Clear the TDRE, RDRF, TEND, PER, FER, and ORER flags in the SCSSR register by reading their values as 1 and then writing 0 to them. Furthermore, exit from interrupt handlers should follow checking to ensure that the corresponding bit has been cleared.

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 communication 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.

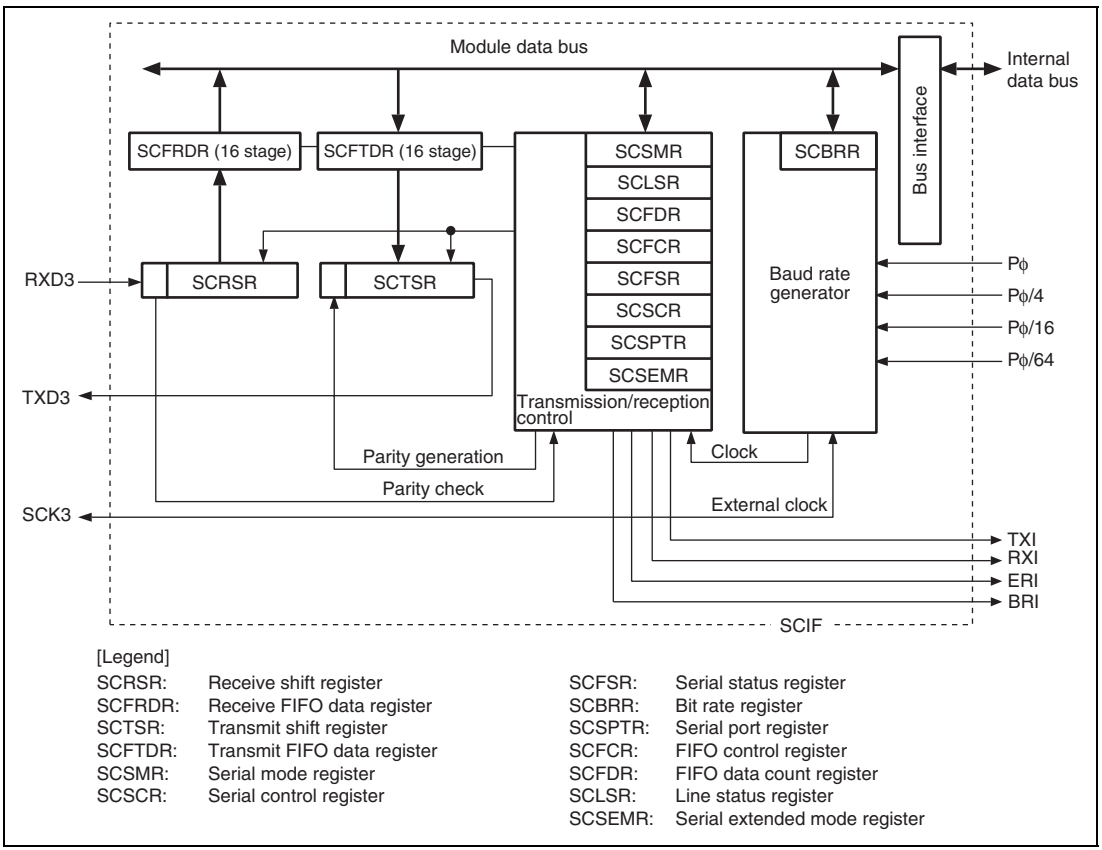


Figure 17.1 Block Diagram of SCIF

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 16-stage FIFO 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 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 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)

	Pϕ (MHz)					
	10*			12*		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	212	0.03
150	2	129	0.16	2	155	0.16
300	2	64	0.16	2	77	0.16
600	1	129	0.16	1	155	0.16
1200	1	64	0.16	1	77	0.16
2400	0	129	0.16	0	155	0.16
4800	0	64	0.16	0	77	0.16
9600	0	32	–1.36	0	38	0.16
19200	0	15	1.73	0	19	0.16
31250	0	9	0.00	0	11	0.00
38400	0	7	1.73	0	9	–2.34

Note: * Cannot be set for this LSI.

	Pϕ (MHz)											
	12.288*			14.7456*			16*			19.6608*		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	3	64	0.70	3	70	0.03	3	86	0.31
150	2	159	0.00	2	191	0.00	2	207	0.16	2	255	0.00
300	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00
600	1	159	0.00	1	191	0.00	1	207	0.16	1	255	0.00
1200	1	79	0.00	1	95	0.00	1	103	0.16	1	127	0.00
2400	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00
31250	0	11	2.40	0	14	–1.70	0	15	0.00	0	19	–1.70
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00

Note: * Cannot be set for this LSI.

	Pϕ (MHz)											
	20			24			24.576			28.7*		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	106	–0.44	3	108	0.08	3	126	0.31
150	3	64	0.16	3	77	0.16	3	79	0.00	3	92	0.46
300	2	129	0.16	2	155	0.16	2	159	0.00	2	186	–0.08
600	2	64	0.16	2	77	0.16	2	79	0.00	2	92	0.46
1200	1	129	0.16	1	155	0.16	1	159	0.00	1	186	–0.08
2400	1	64	0.16	1	77	0.16	1	79	0.00	1	92	0.46
4800	0	129	0.16	0	155	0.16	0	159	0.00	0	186	–0.08
9600	0	64	0.16	0	77	0.16	0	79	0.00	0	92	0.46
19200	0	32	–1.36	0	38	0.16	0	39	0.00	0	46	–0.61
31250	0	19	0.00	0	23	0.00	0	24	–1.70	0	28	–1.03
38400	0	15	1.73	0	19	–2.34	0	19	0.00	0	22	1.55

Note: * Cannot be set for this LSI.

	Pϕ (MHz)											
	30*[1]			33*[1]			40			50*[2]		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	145	0.33	3	117	–0.25	3	221	–0.02
150	3	97	–0.35	3	106	0.39	3	129	0.16	3	162	–0.15
300	2	194	0.16	2	214	–0.07	3	64	0.16	3	80	0.47
600	2	97	–0.35	2	106	0.39	2	129	0.16	2	162	–0.15
1200	1	194	0.16	1	214	–0.07	2	64	0.16	2	80	0.47
2400	1	97	–0.35	1	106	0.39	1	129	0.16	1	162	–0.15
4800	0	194	–1.36	0	214	–0.07	1	64	0.16	1	80	0.47
9600	0	97	–0.35	0	106	0.39	0	129	0.16	0	162	–0.15
19200	0	48	–0.35	0	53	–0.54	0	64	0.16	0	80	–0.47
31250	0	29	0.00	0	32	0.00	0	39	0.00	0	49	0
38400	0	23	1.73	0	26	–0.54	0	32	–1.36	0	40	–0.76

Notes: Settings with an error of 1% or less are recommended.

1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

Table 17.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode)

Bit Rate (bit/s)	Pϕ (MHz)											
	16*[1]		28.7*[1]		30*[1]		33*[1]		40		50*[2]	
	n	N	n	N	n	N	n	N	n	N	n	N
110												
250	3	249										
500	3	124	3	223	3	233	3	255	—	—	—	—
1 k	2	249	3	111	3	116	3	125	3	152	3	194
2.5 k	2	99	2	178	2	187	2	200	2	243	3	77
5 k	1	199	2	89	2	93	2	100	2	121	2	155
10 k	1	99	1	178	1	187	1	200	2	60	2	77
25 k	0	159	1	71	1	74	1	80	1	97	1	124
50 k	0	79	0	143	0	149	0	160	1	48	0	249
100 k	0	39	0	71	0	74	0	80	0	97	0	124
250 k	0	15	—	—	0	29	0	31	0	38	0	49
500 k	0	7	—	—	0	14	0	15	0	19	0	24
1 M	0	3					0	7	0	9	—	—
2 M	0	1							0	3	—	—

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

Table 17.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 17.7 and 17.8 list the maximum bit rates when the external clock input is used.

Table 17.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
12*[1]	375000	0	0
14.7456*[1]	460800	0	0
16*[1]	500000	0	0
19.6608*[1]	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7*[1]	896875	0	0
30*[1]	937500	0	0
33*[1]	1031250	0	0
40	1250000	0	0
50*[2]	1562500	0	0

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

Table 17.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
12*[1]	3.0000	187500
14.7456*[1]	3.6864	230400
16*[1]	4.0000	250000
19.6608*[1]	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7*[1]	7.1750	448436
30*[1]	7.5000	468750
33*[1]	8.2500	515625
40	10.0000	625000
50*[2]	12.5000	781250

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

Table 17.8 Maximum Bit Rates with External Clock Input

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
16*[1]	2.6667	2666666.7
24	4.0000	4000000.0
28.7	4.7833	4783333.3
30*[1]	5.0000	5000000.0
33*[1]	5.5000	5500000.0
40	6.6667	6666666.7
50*[2]	8.0000	8000000.0

Notes: 1. Cannot be set for this LSI.
2. This is only available for the SH7239B and SH7237B.

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 17.9. 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 17.10.

(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 17.9 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 17.10 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR				
Bit 7	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	Mode	Clock 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.

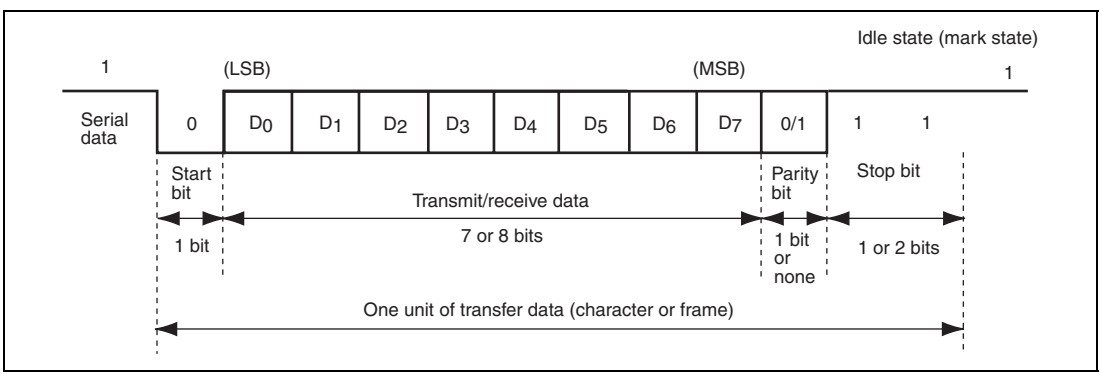


Figure 17.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 17.11 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 17.11 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 17.10.

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.

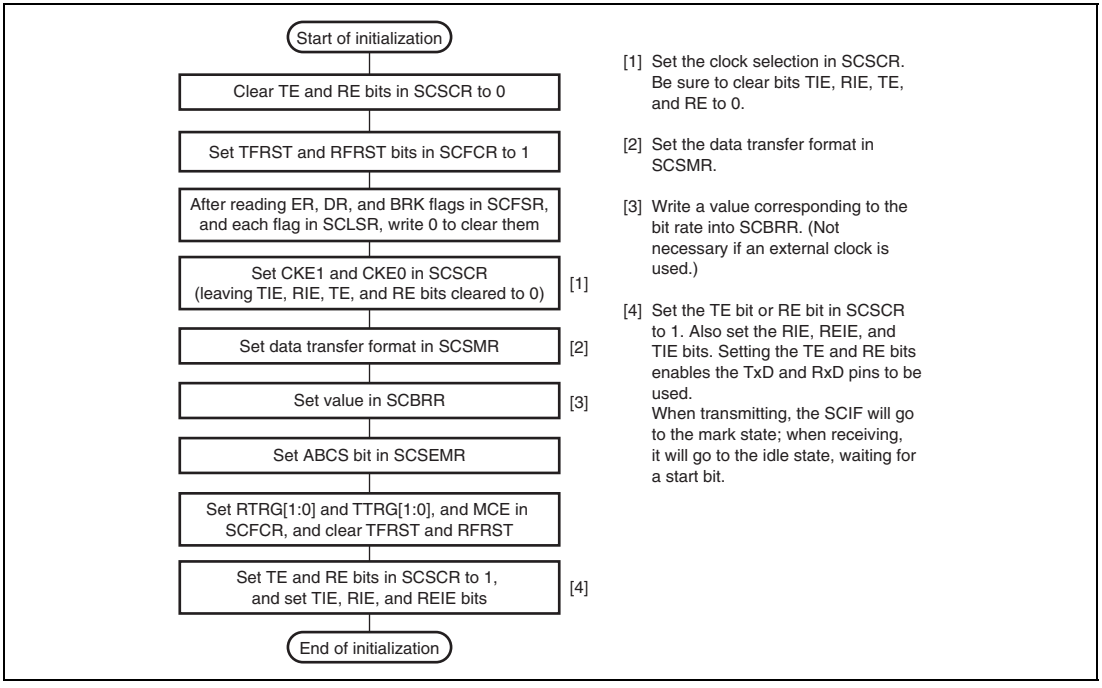


Figure 17.3 Sample Flowchart for SCIF Initialization

- 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.

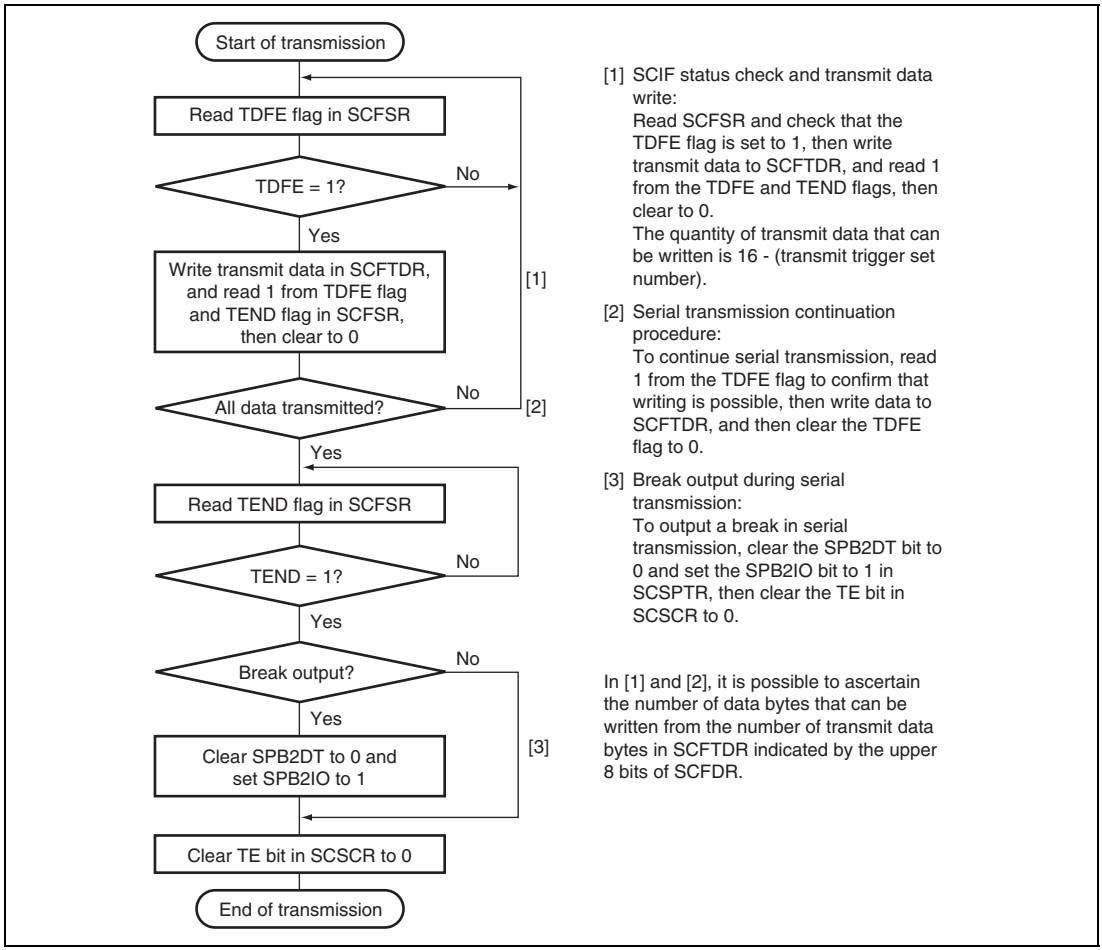


Figure 17.4 Sample Flowchart for Transmitting Serial Data

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.

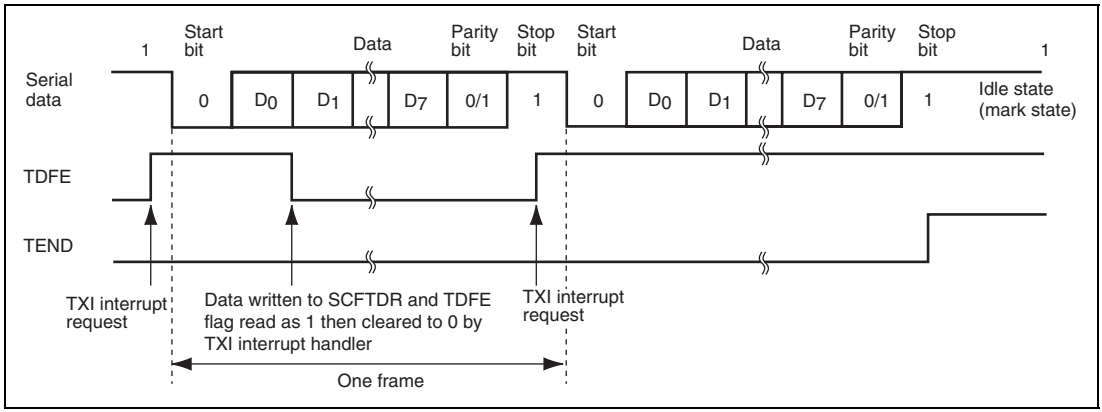


Figure 17.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

- 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.

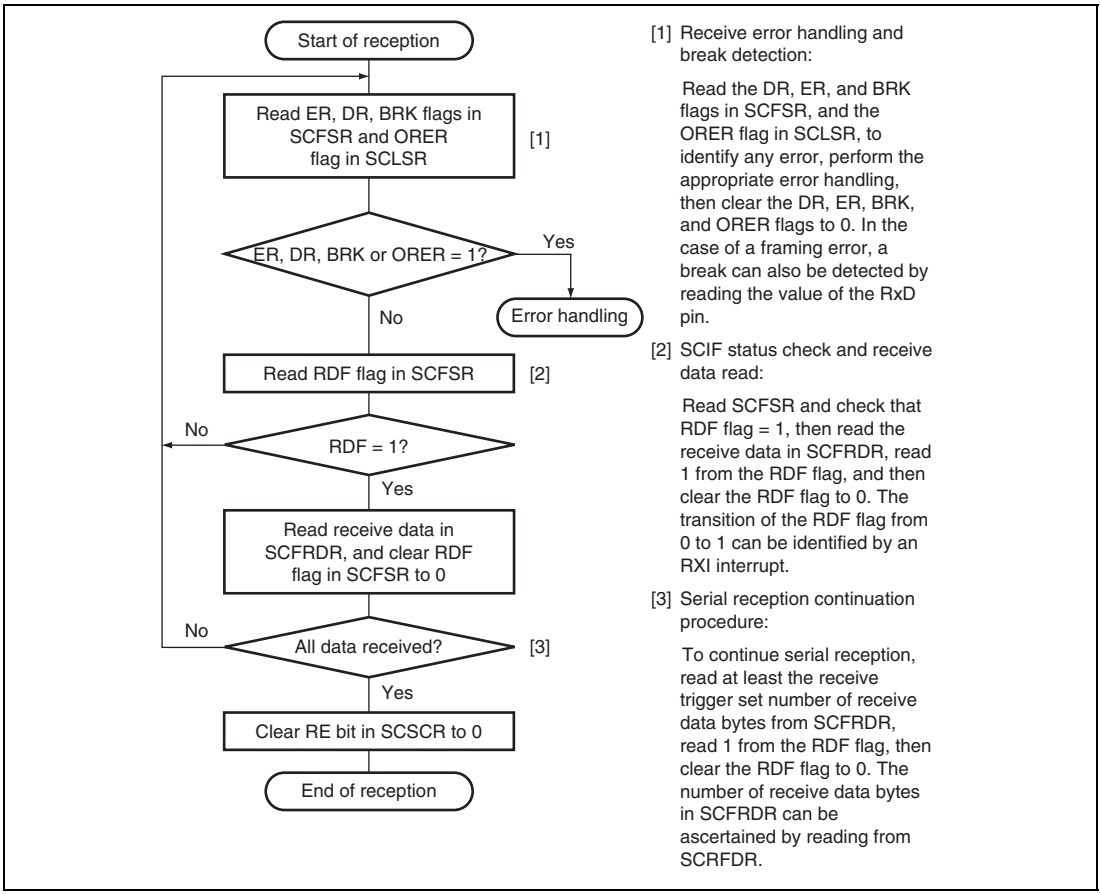
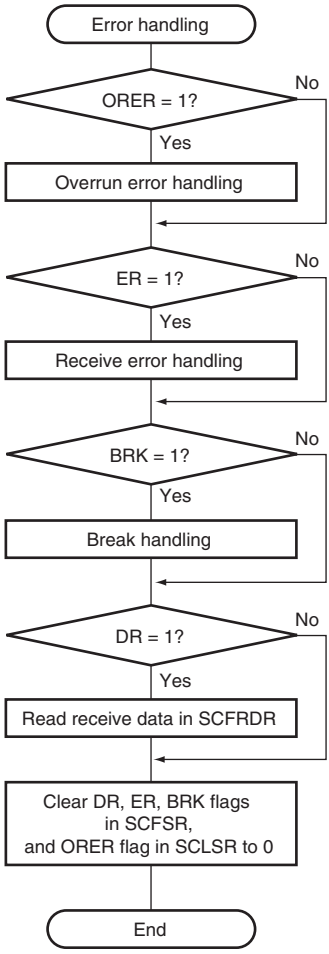


Figure 17.6 Sample Flowchart for Receiving Serial Data



- 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.

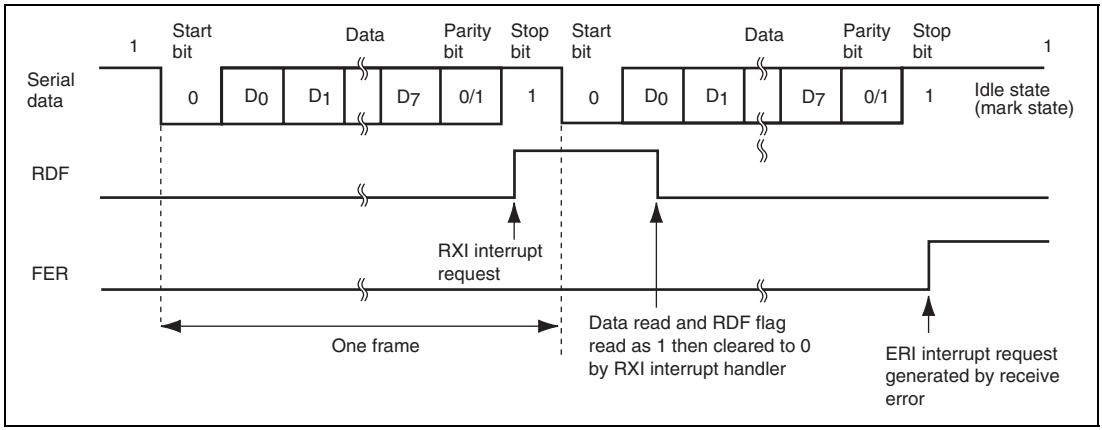


Figure 17.8 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

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.

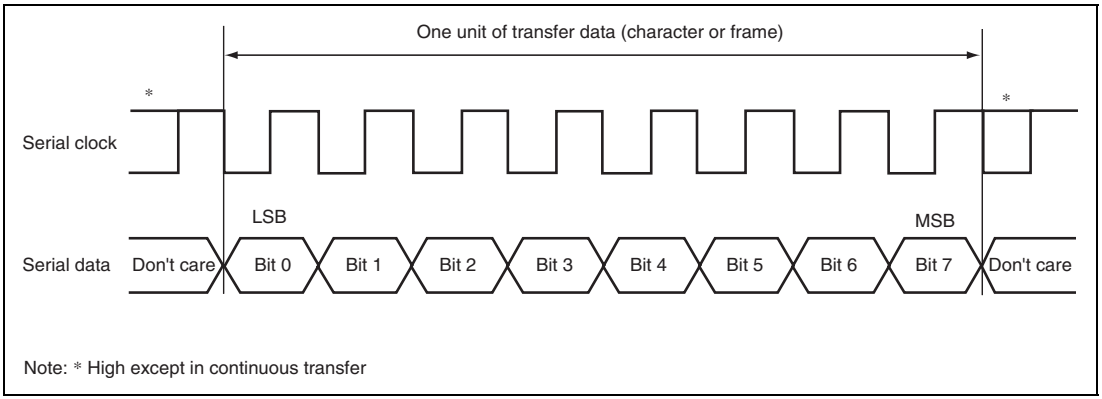


Figure 17.9 Data Format in Clocked Synchronous Communication

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.

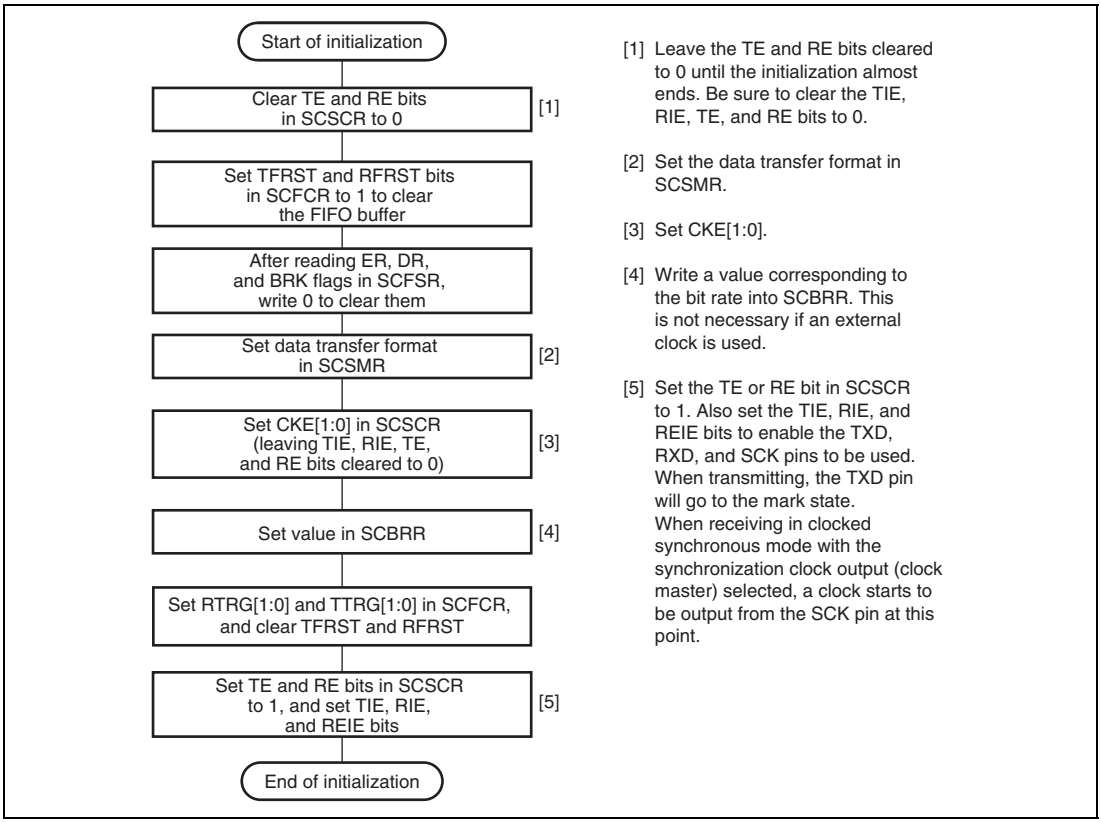


Figure 17.10 Sample Flowchart for SCIF Initialization

- 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.

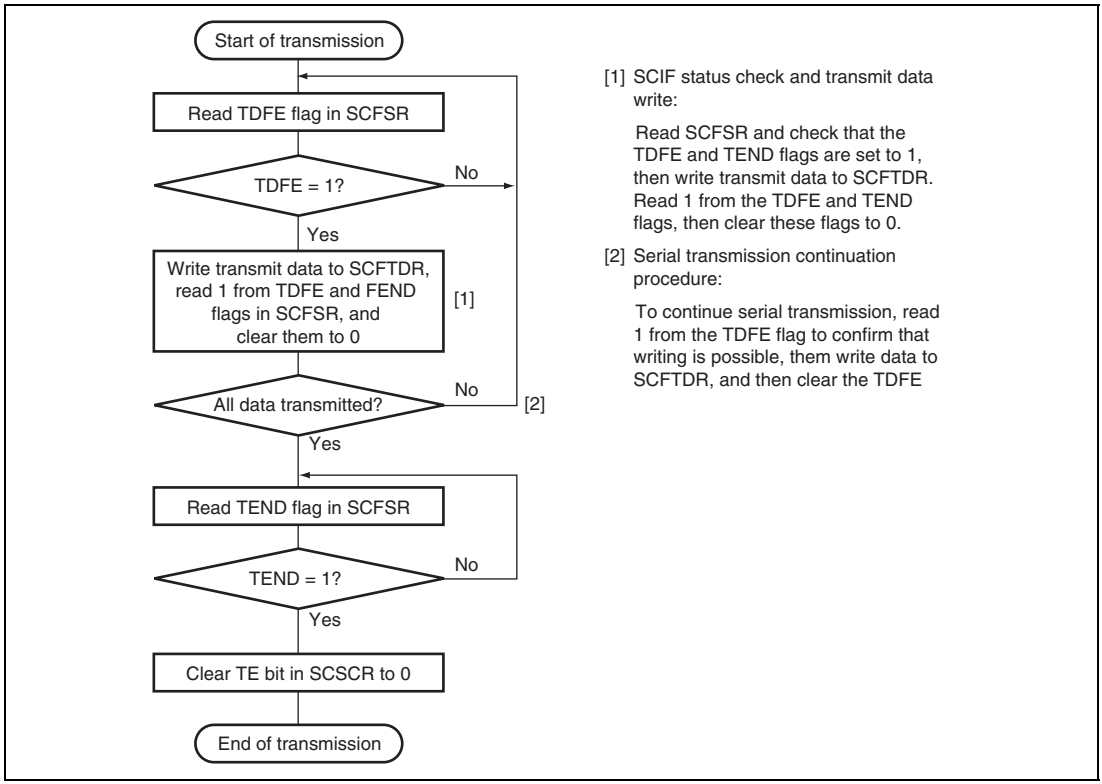


Figure 17.11 Sample Flowchart for Transmitting Serial Data

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.

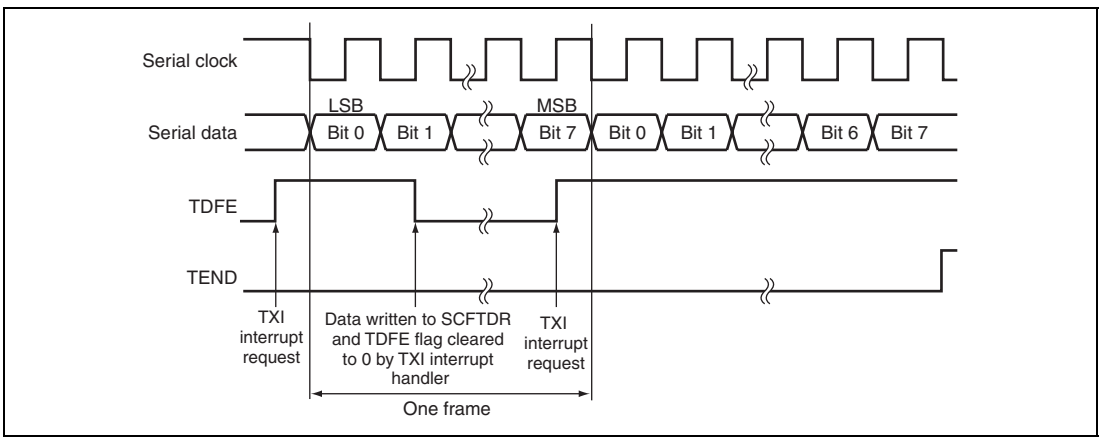


Figure 17.12 Example of SCIF Transmit Operation

- 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.

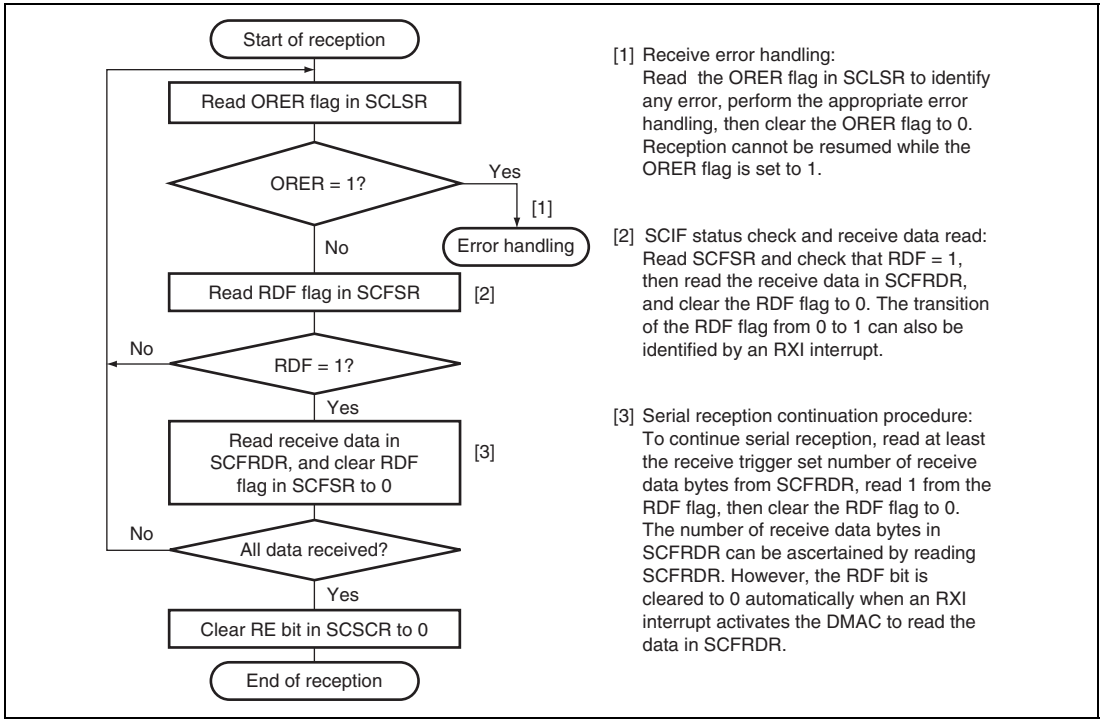


Figure 17.13 Sample Flowchart for Receiving Serial Data (1)

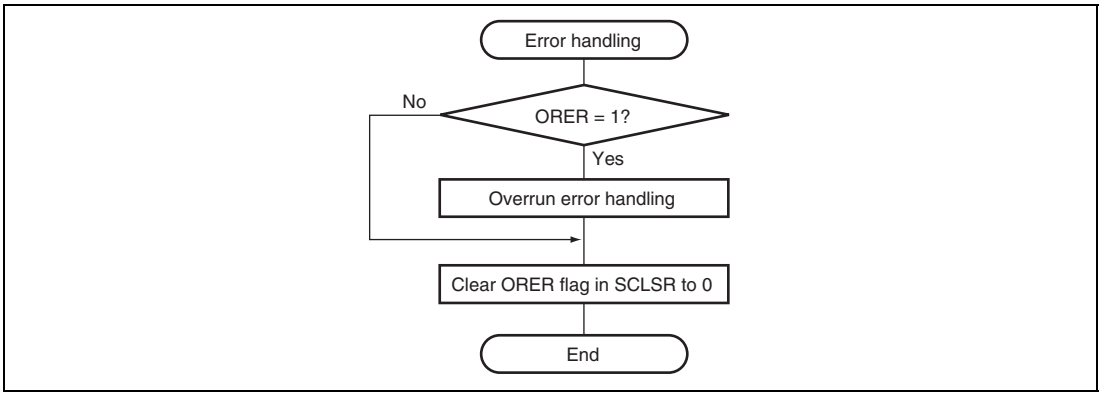


Figure 17.14 Sample Flowchart for Receiving Serial Data (2)

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.

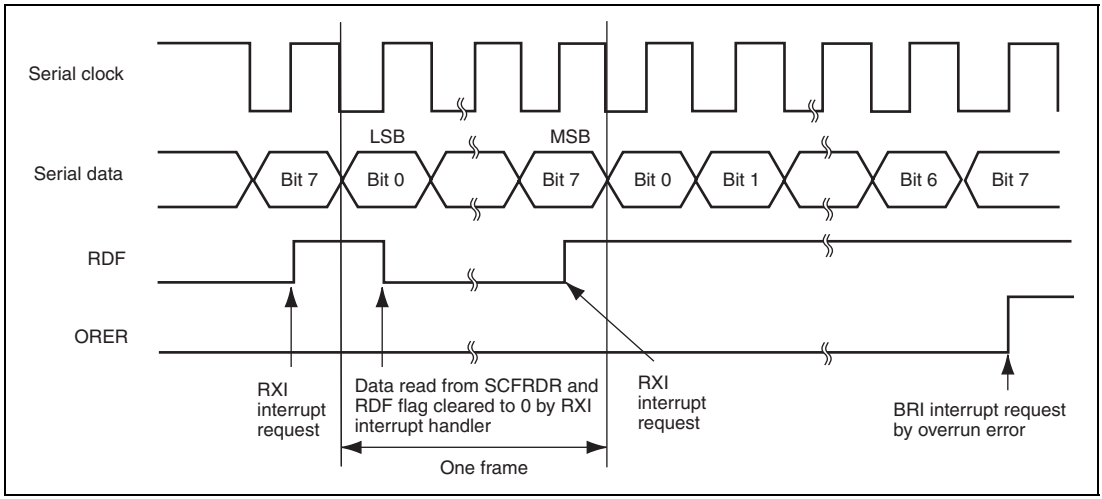


Figure 17.15 Example of SCIF Receive Operation

- 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.

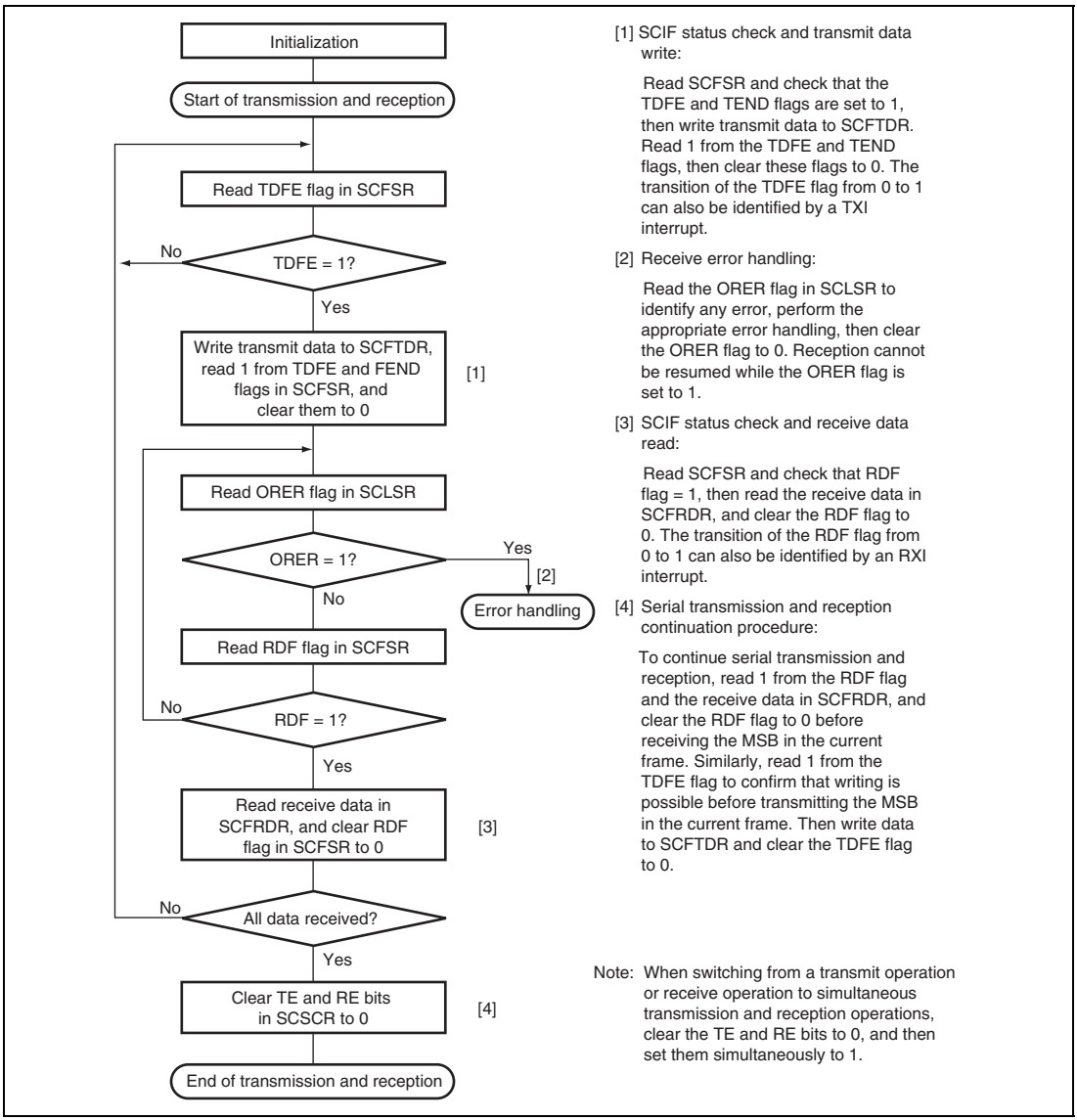


Figure 17.16 Sample Flowchart for Transmitting/Receiving Serial Data

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 17.12 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 17.12 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.

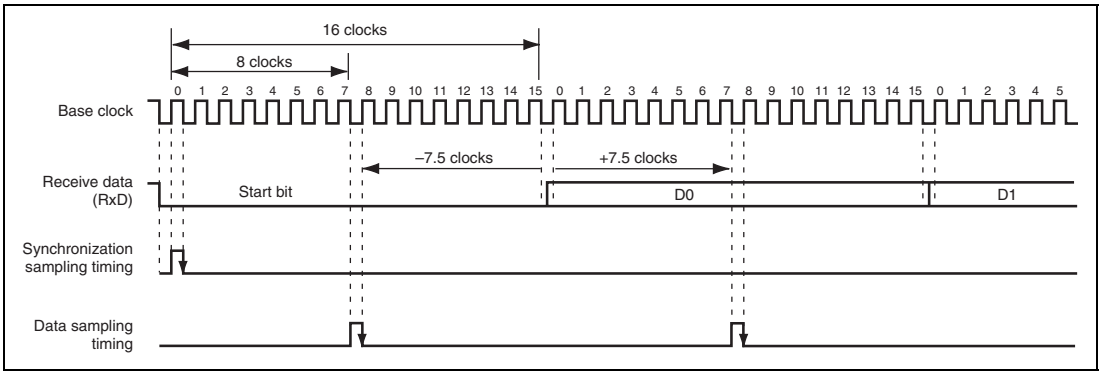


Figure 17.17 Receive Data Sampling Timing in Asynchronous Mode

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.

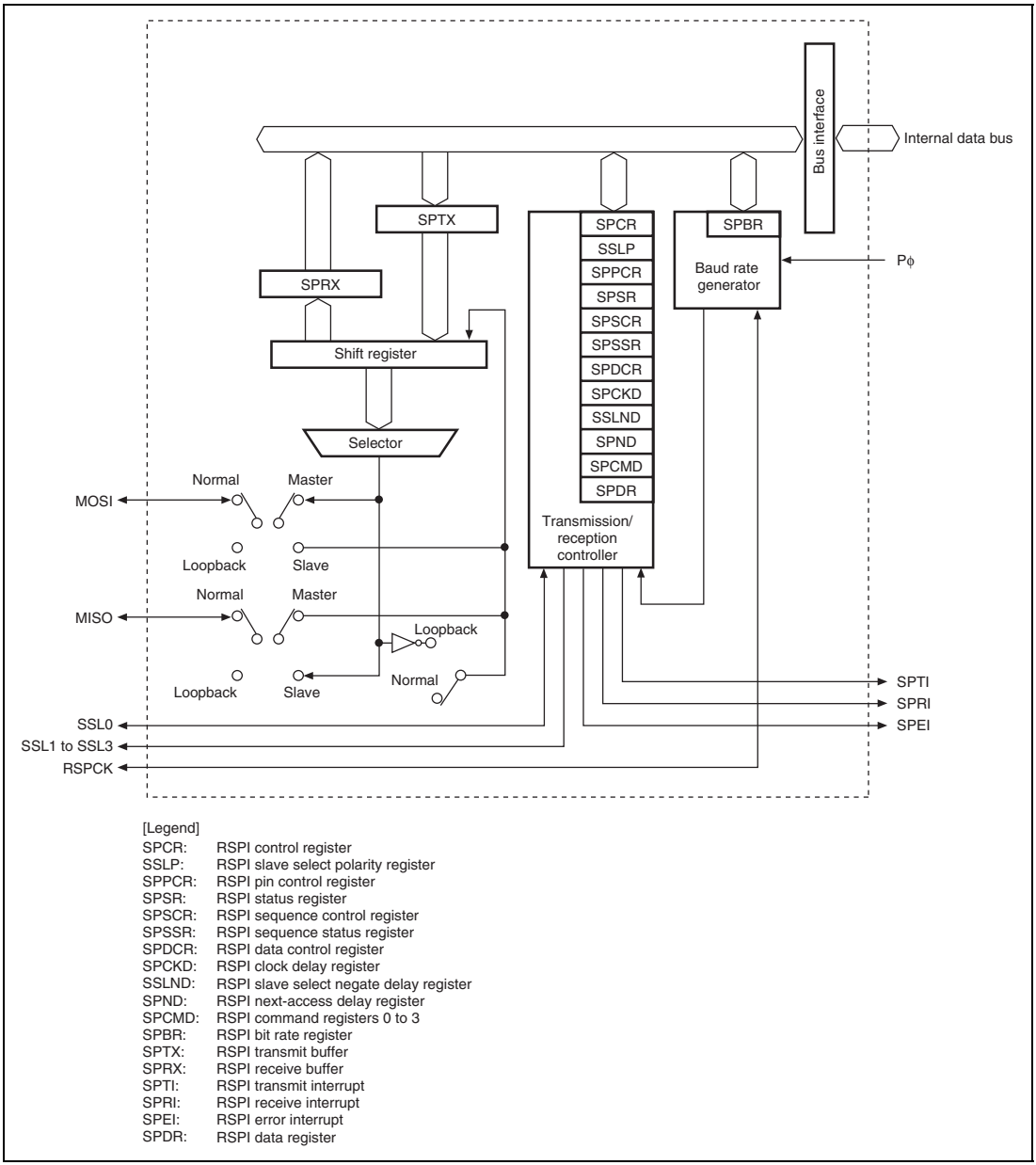
7. Interrupt Sources

- 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.

Figure 18.1 shows an RSPI block diagram for one channel. When the CPU accesses the RSPI control registers, a peripheral bus (P-bus) is used.



[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 (for One Channel)

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	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 to 0	SSL[3:0]P	0000	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

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 1: Valid data found 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 [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 1: No data 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). [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 1: A mode fault error occurs [Clearing conditions] • Power-on reset • When 0 is written in MODF after reading MODF = 1. [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. 0: RSPI transfers the data. 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. [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 1: An overrun error occurs [Clearing conditions] • Power-on reset • When 0 is written in OVRF after reading OVRF = 1. [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.

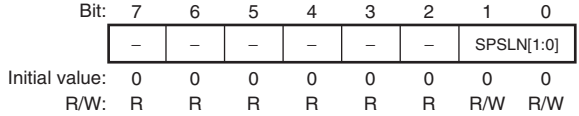
In the normal operating method, the CPU, DTC, and DMAC read the receive buffer when the SPRF bit in SPSR is 1 (a condition in which unread data is stored in the receive buffer). When the SPRF or OVRF bit in SPSR is 1, the RSPI does not update the receive buffer of SPDR at the end of a serial transfer.

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	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 SPSLN[1:0] 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.

	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	—	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	—	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]	00	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 the BRDV[1:0] bits in the RSPI command registers (SPCMD0 to SPCMD3).

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 SPCMD3). 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
5	2	48	333 kbps	417 kbps	667 kbps	833 kbps	1.04 kbps
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

Notes: —: Setting prohibited
*: Only for SH7239B and SH7237B

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.
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.

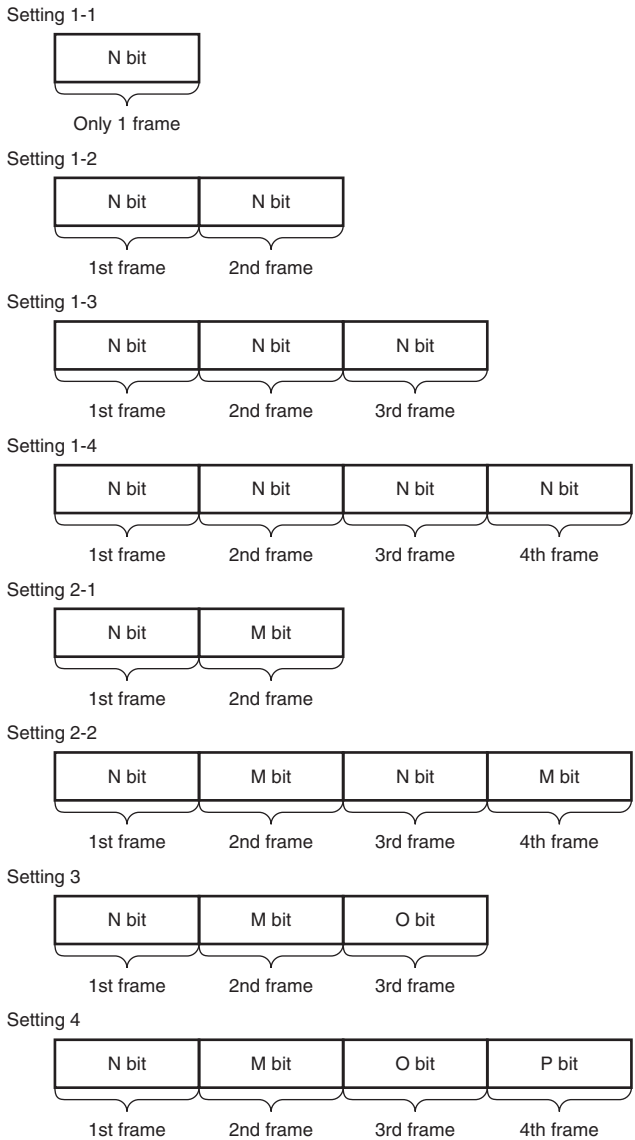
Bit	Bit Name	Initial Value	R/W	Description
1, 0	SPFC[1:0]	00	R/W	Frame Count Setting These bits specify the number of frames that can be stored in the SPDR register. Up to 4 frames can be transmitted or received at a time upon transmission or reception activation according to the setting combinations 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.	SPB1 and 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 when the SLNDEN bit in SPCMD is 1. 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

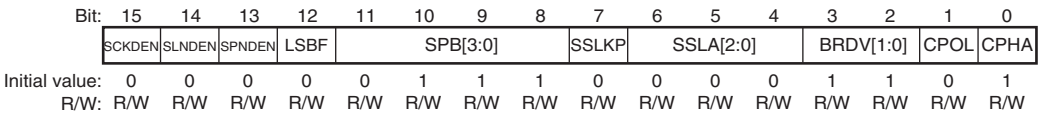
18.3.13 RSPI Command Register (SPCMD)

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
MOSI signal	Input	Output	Output/Hi-Z	Input	Output
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/Hi-Z	Open-drain output/Hi-Z

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 disactive 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.

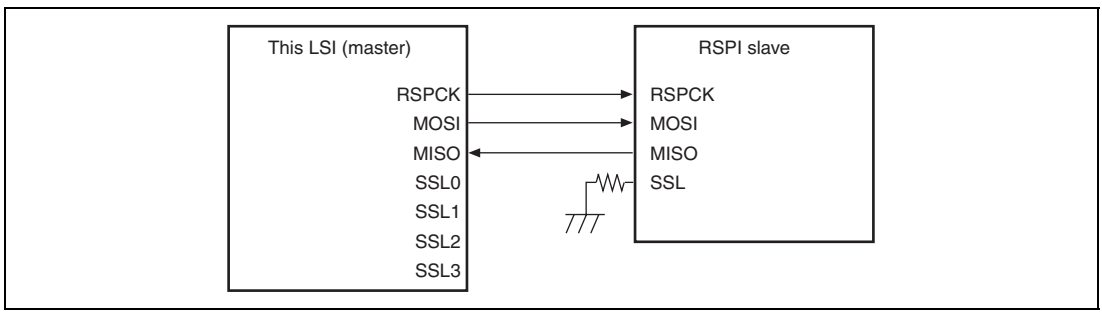


Figure 18.2 Single-Master/Single-Slave Configuration Example (This LSI = Master)

(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 disactive level, the pin state becomes Hi-Z.

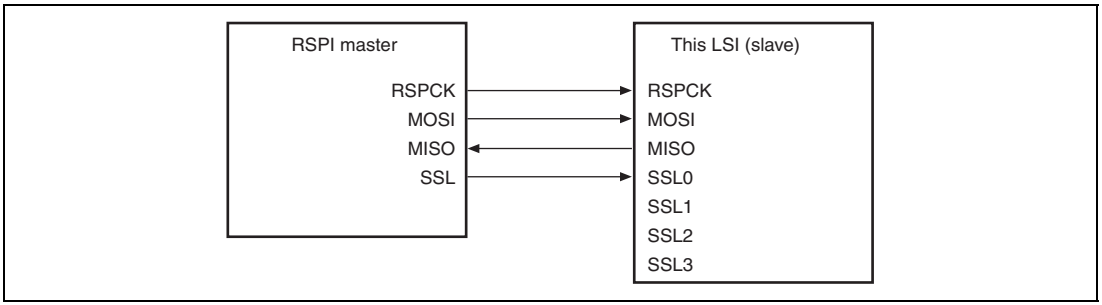


Figure 18.3 Single-Master/Single-Slave Configuration Example (This LSI = Slave)

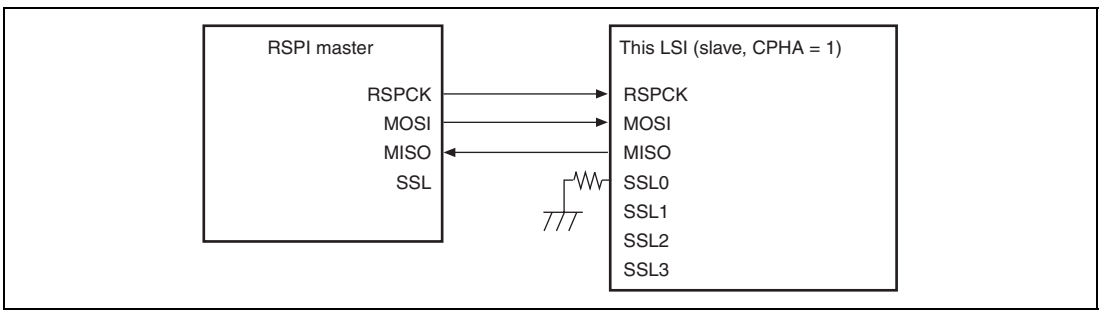


Figure 18.4 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

(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.

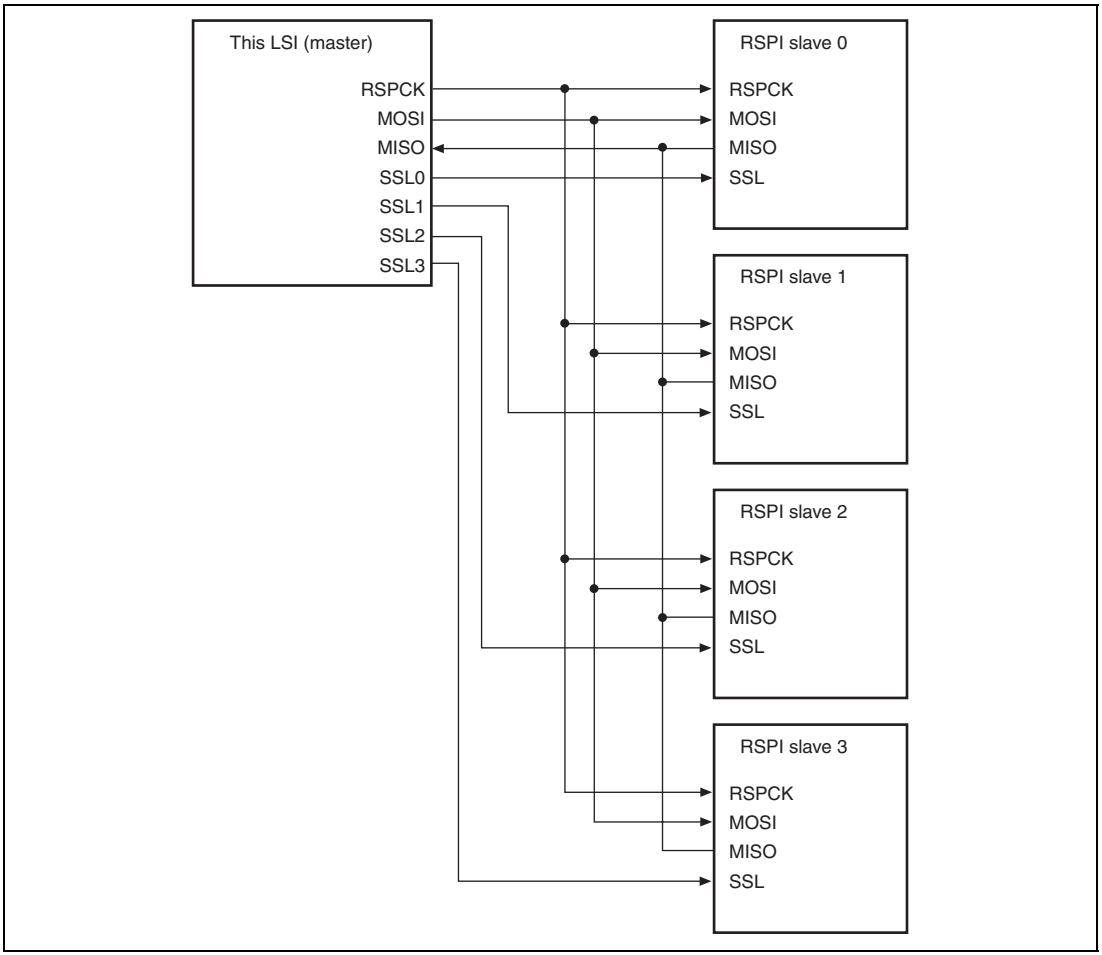


Figure 18.5 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

(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.

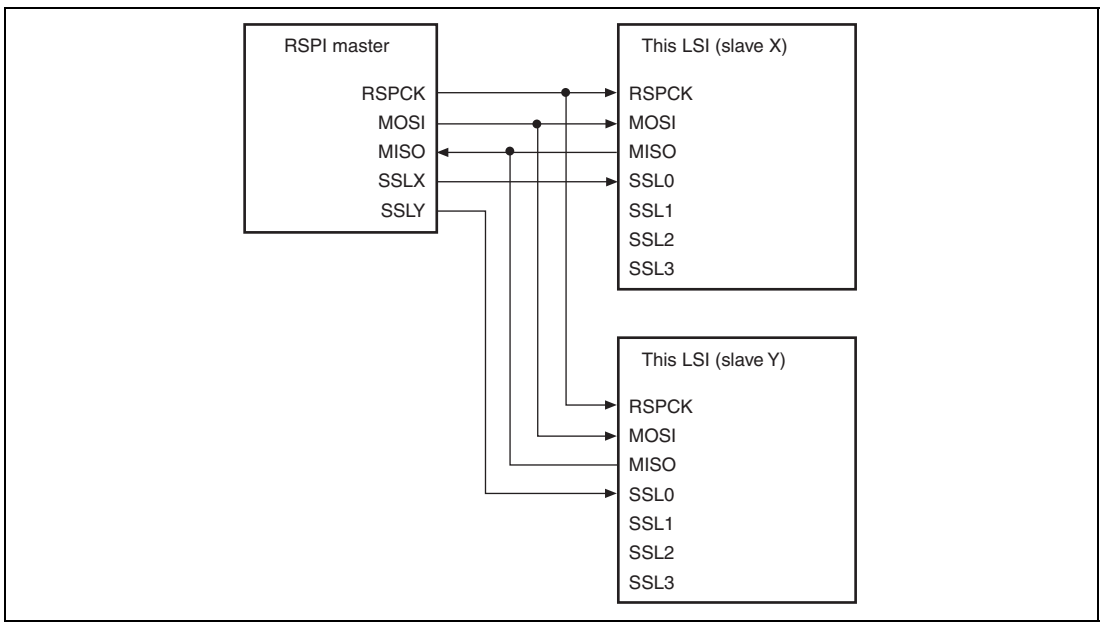


Figure 18.6 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)

(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.

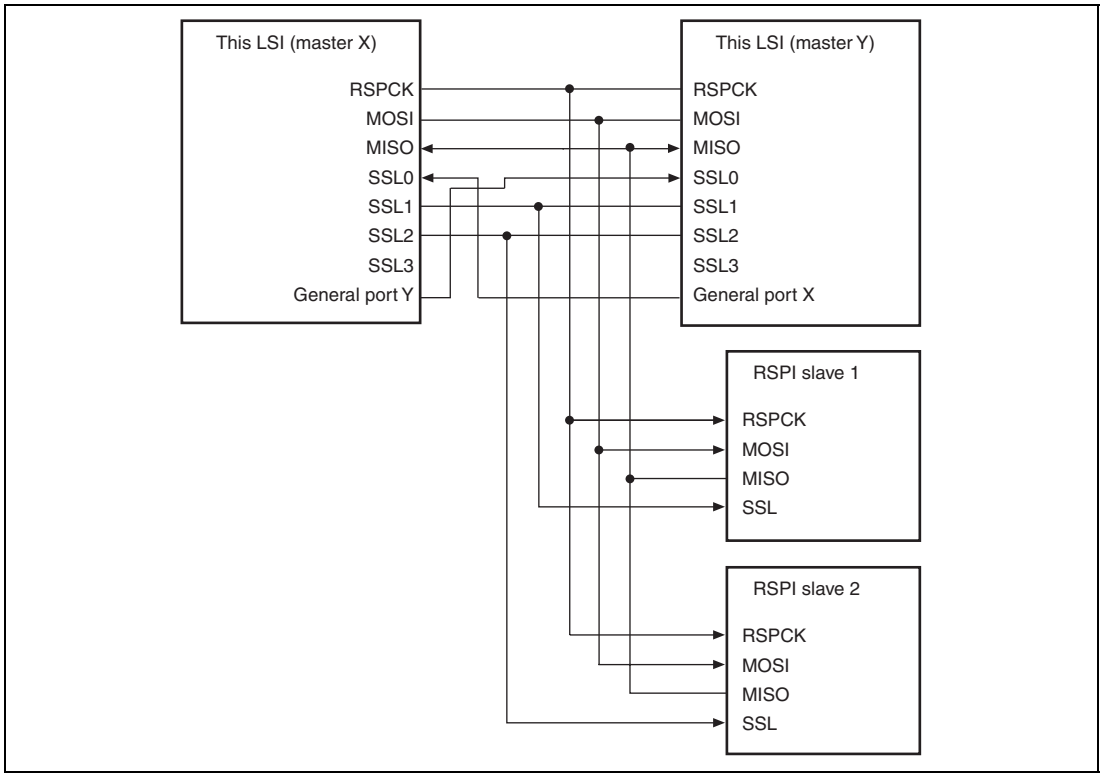


Figure 18.7 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

(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.

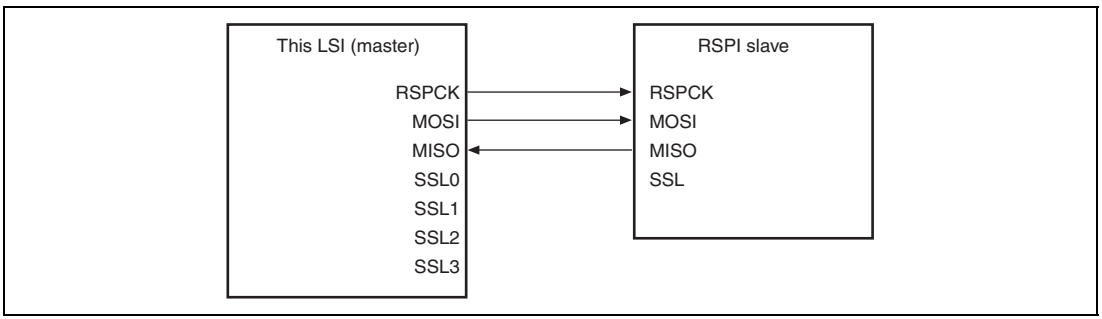


Figure 18.8 Master (Clock Synchronous)/Slave (Clock Synchronous) Configuration Example (This LSI = Master)

(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.

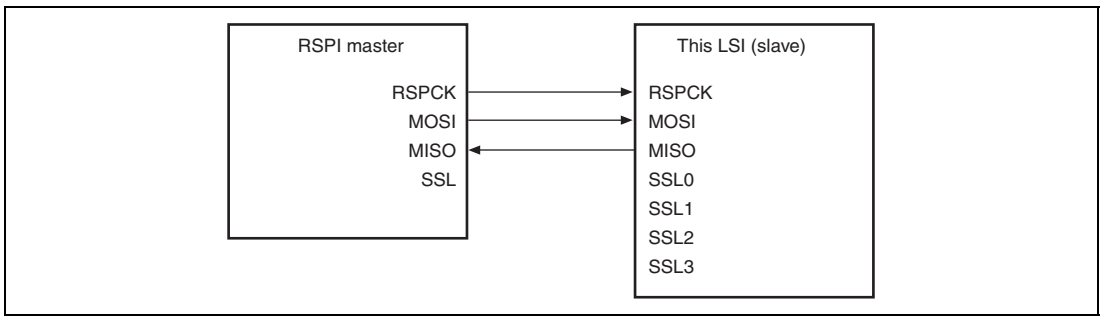


Figure 18.9 Master (Clock Synchronous)/Slave (Clock Synchronous) Configuration Example (This LSI = Slave, CPHA = 1)

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.

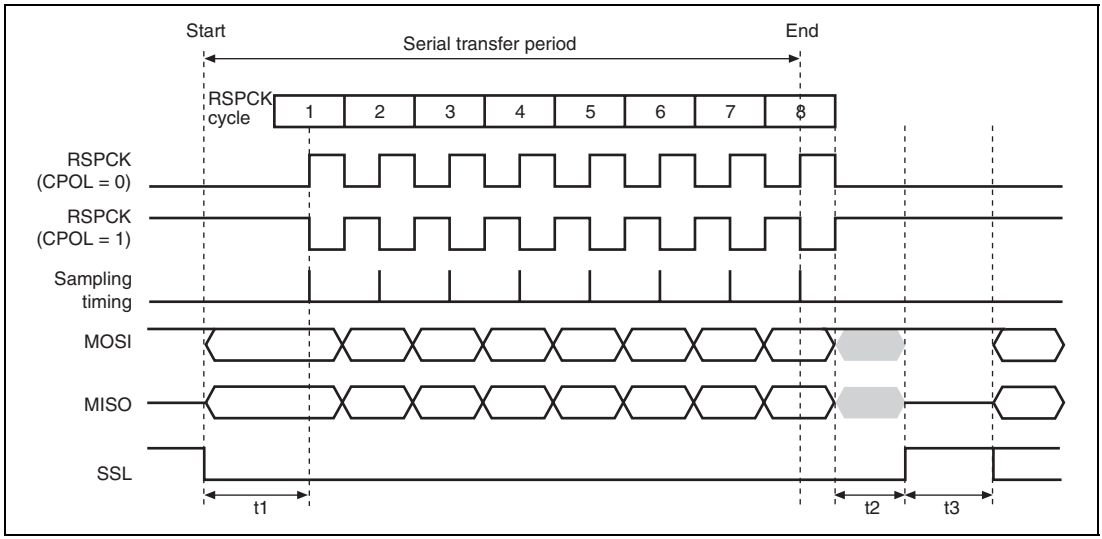


Figure 18.10 RSPI Transfer Format (CPHA = 0)

(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.

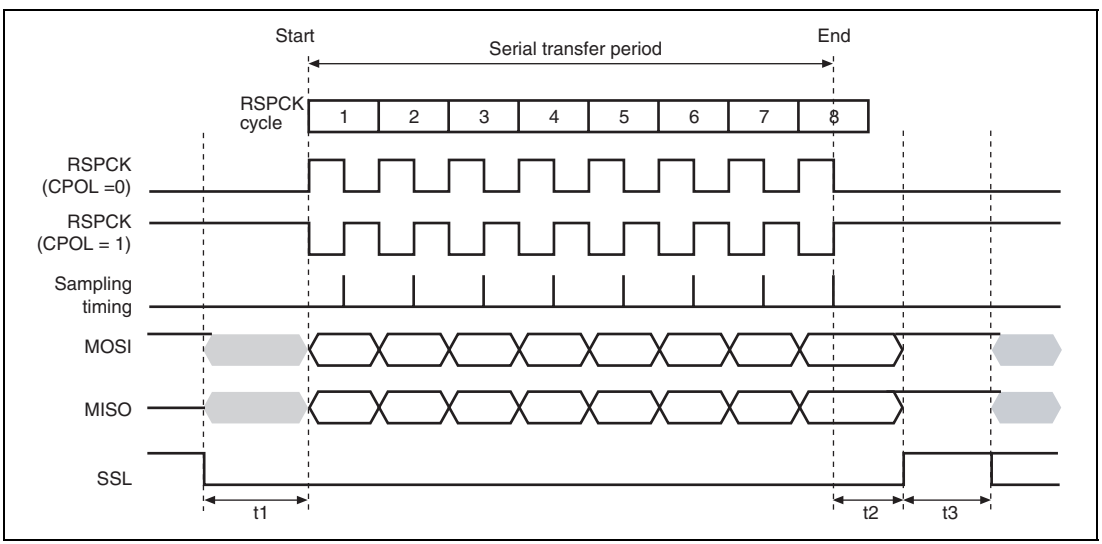


Figure 18.11 RSPI Transfer Format (CPHA = 1)

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.

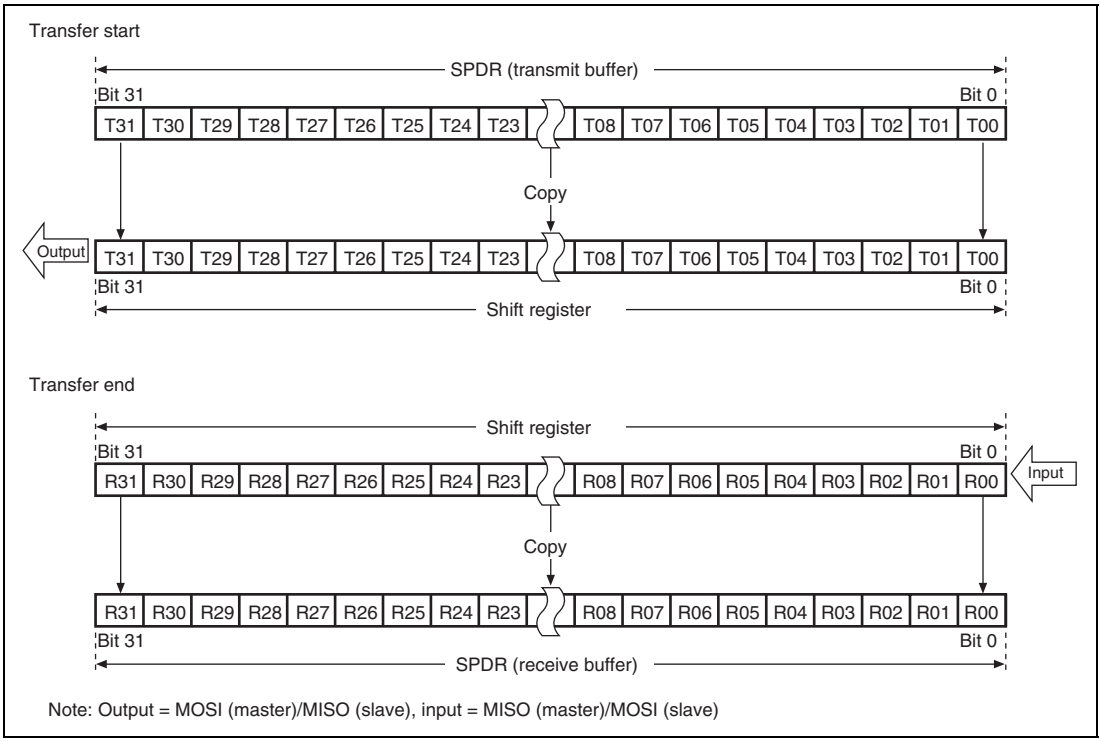


Figure 18.12 MSB First Transfer (32-Bit Data)

(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.

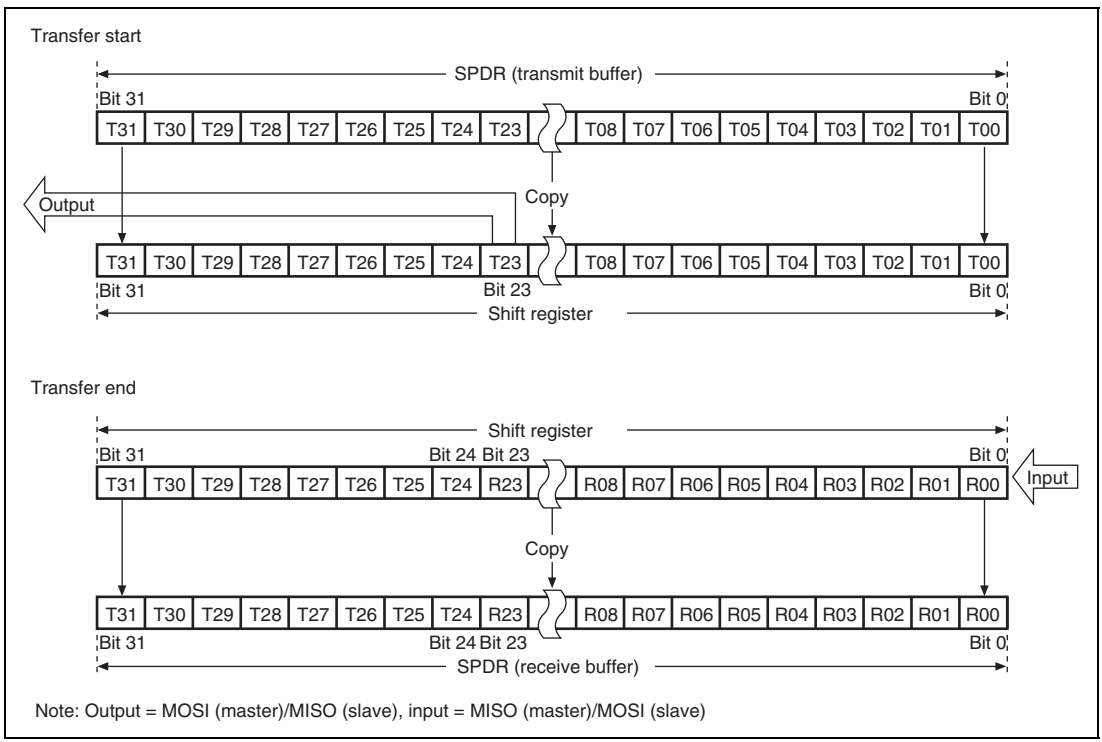


Figure 18.13 MSB First Transfer (24-Bit Data)

(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.

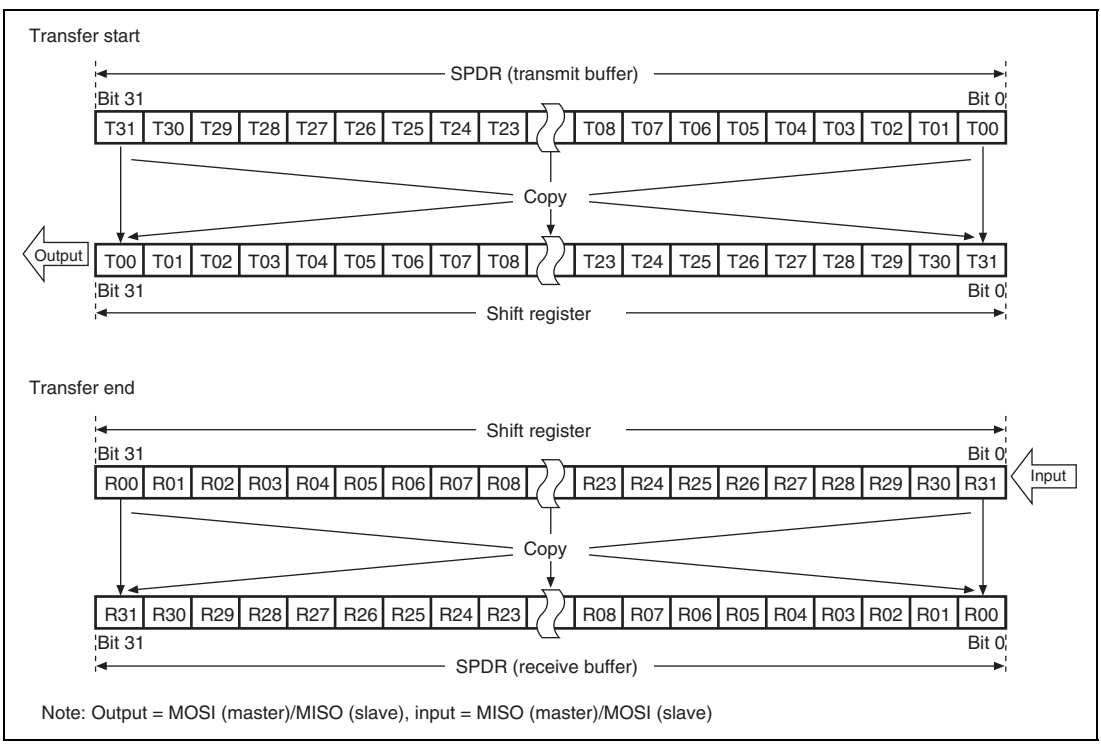


Figure 18.14 LSB First Transfer (32-Bit Data)

(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.

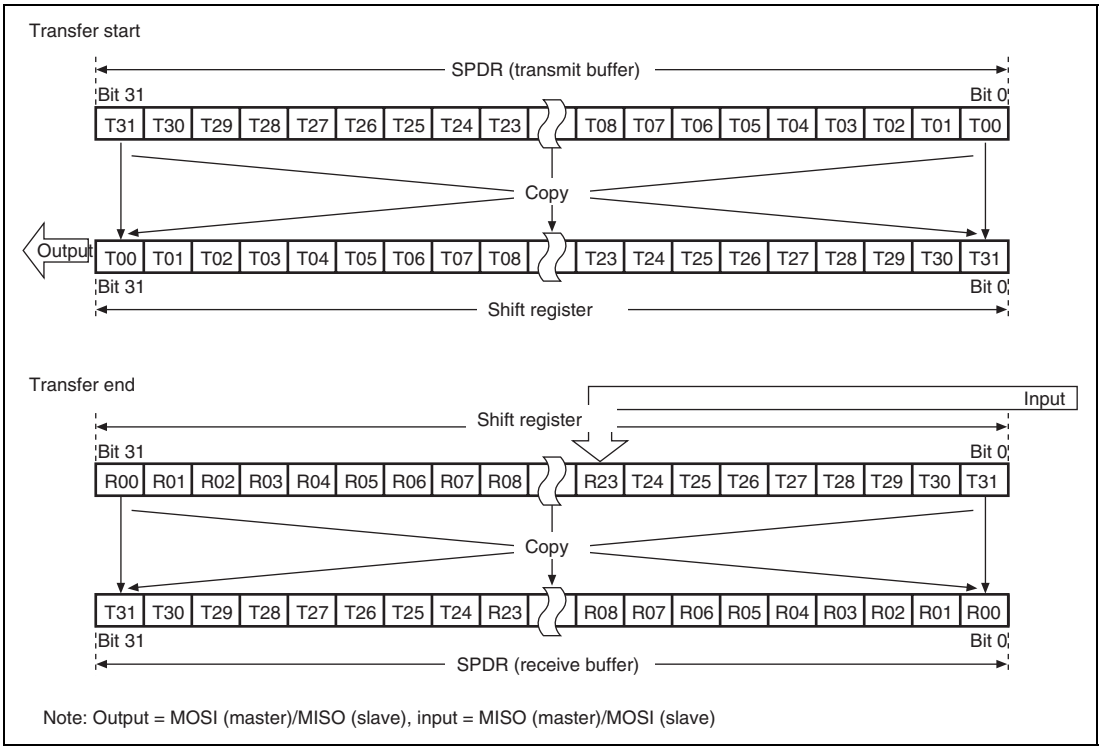


Figure 18.15 LSB First Transfer (24-Bit Data)

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).

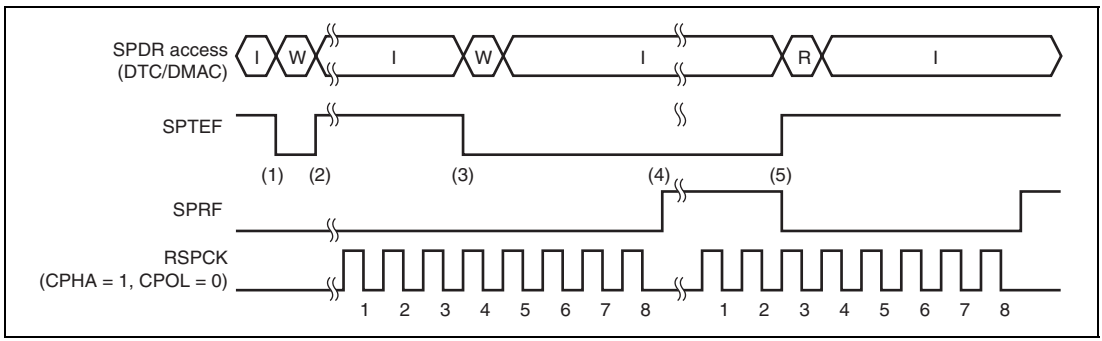


Figure 18.16 SPTEF and SPRF Bit Operation Example

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).

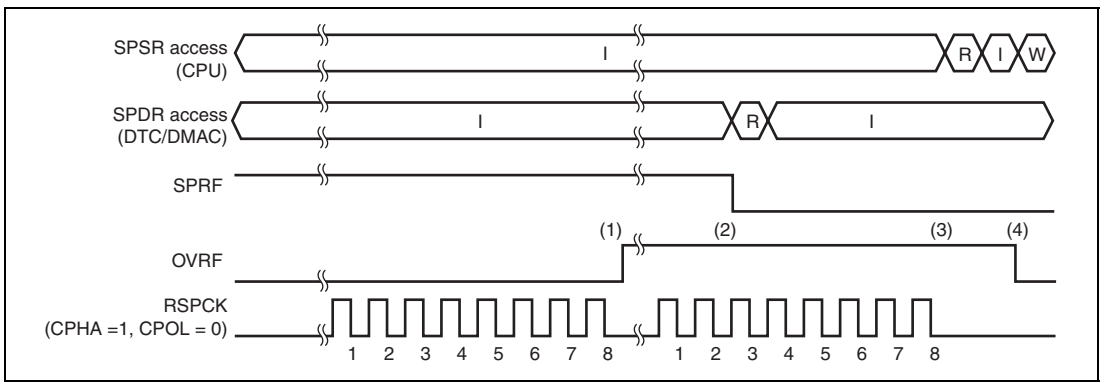


Figure 18.17 SPRF and OVRF Bit Operation Example

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.

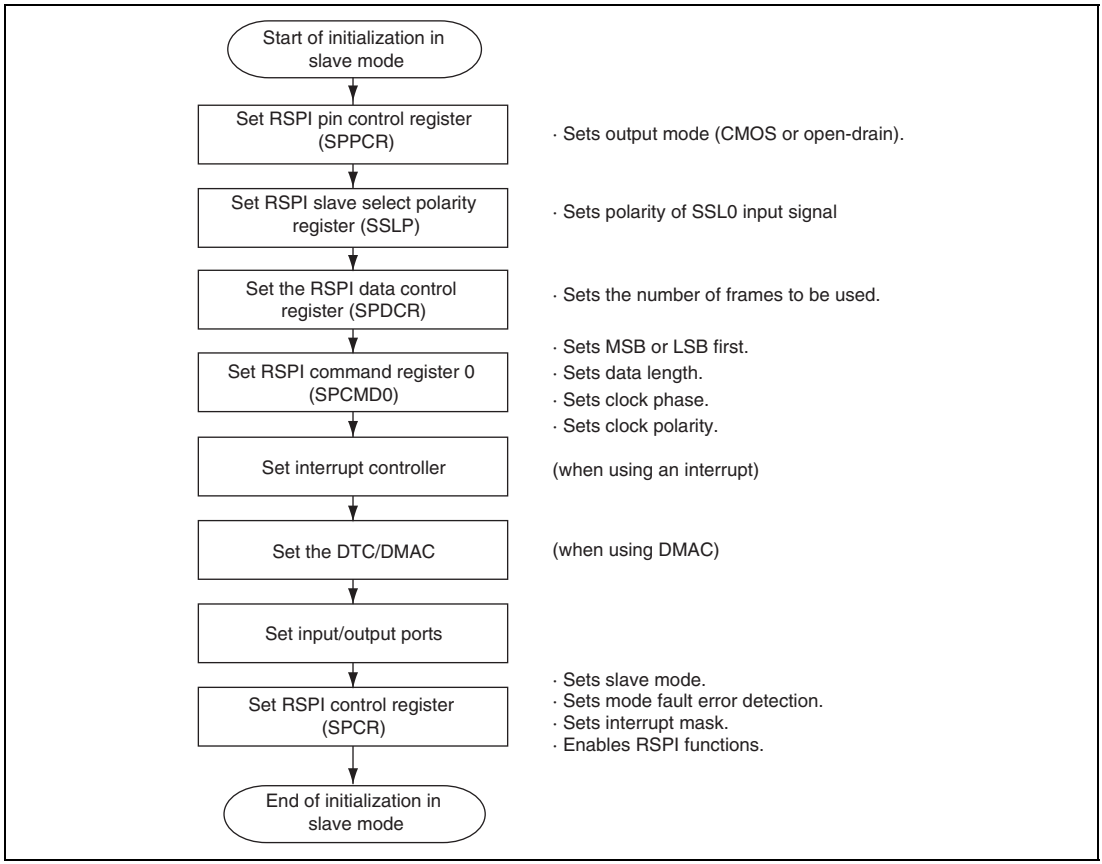


Figure 18.18 Example of Initialization Flowchart in Slave Mode

(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.

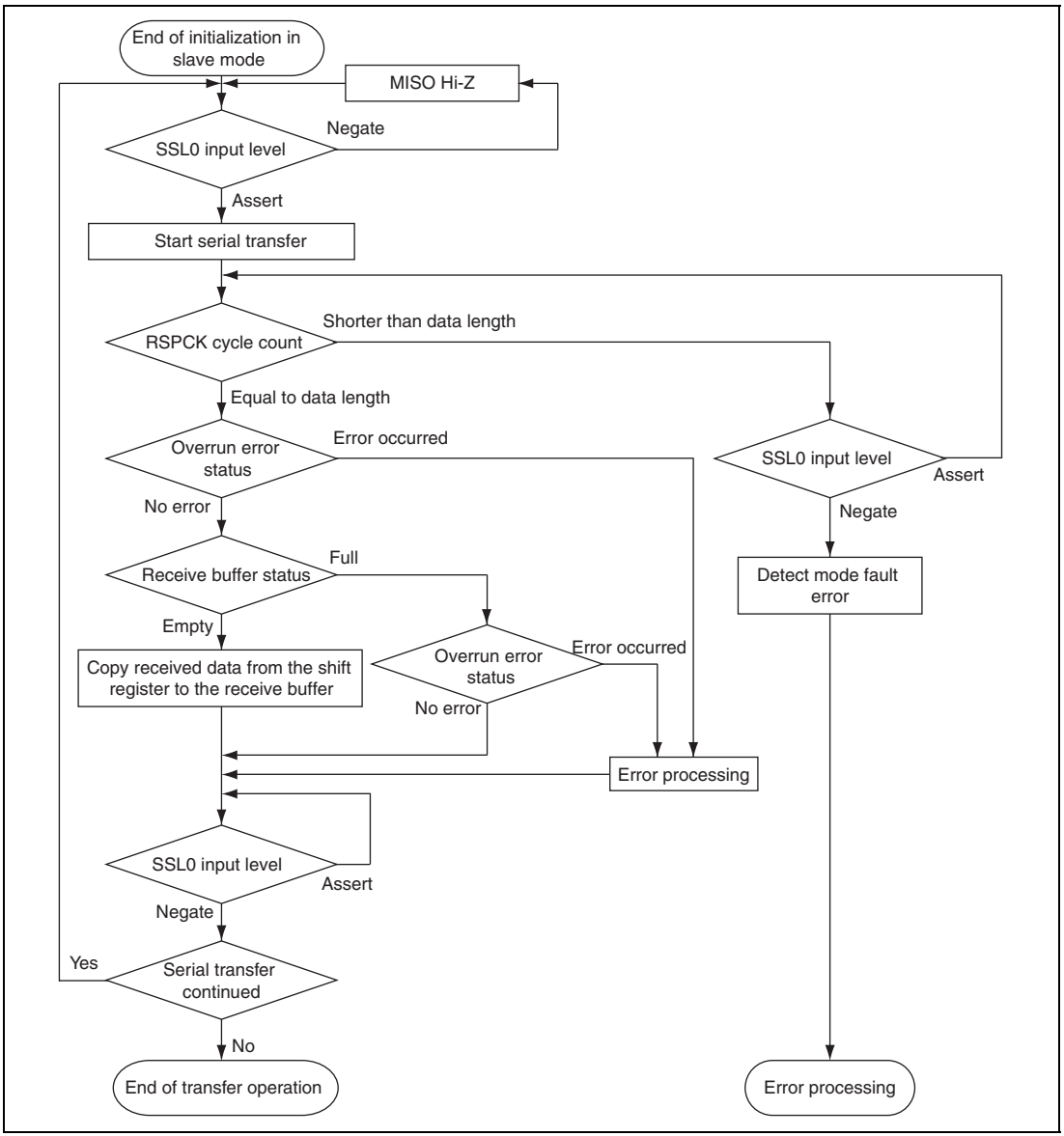


Figure 18.19 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 0)

(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.

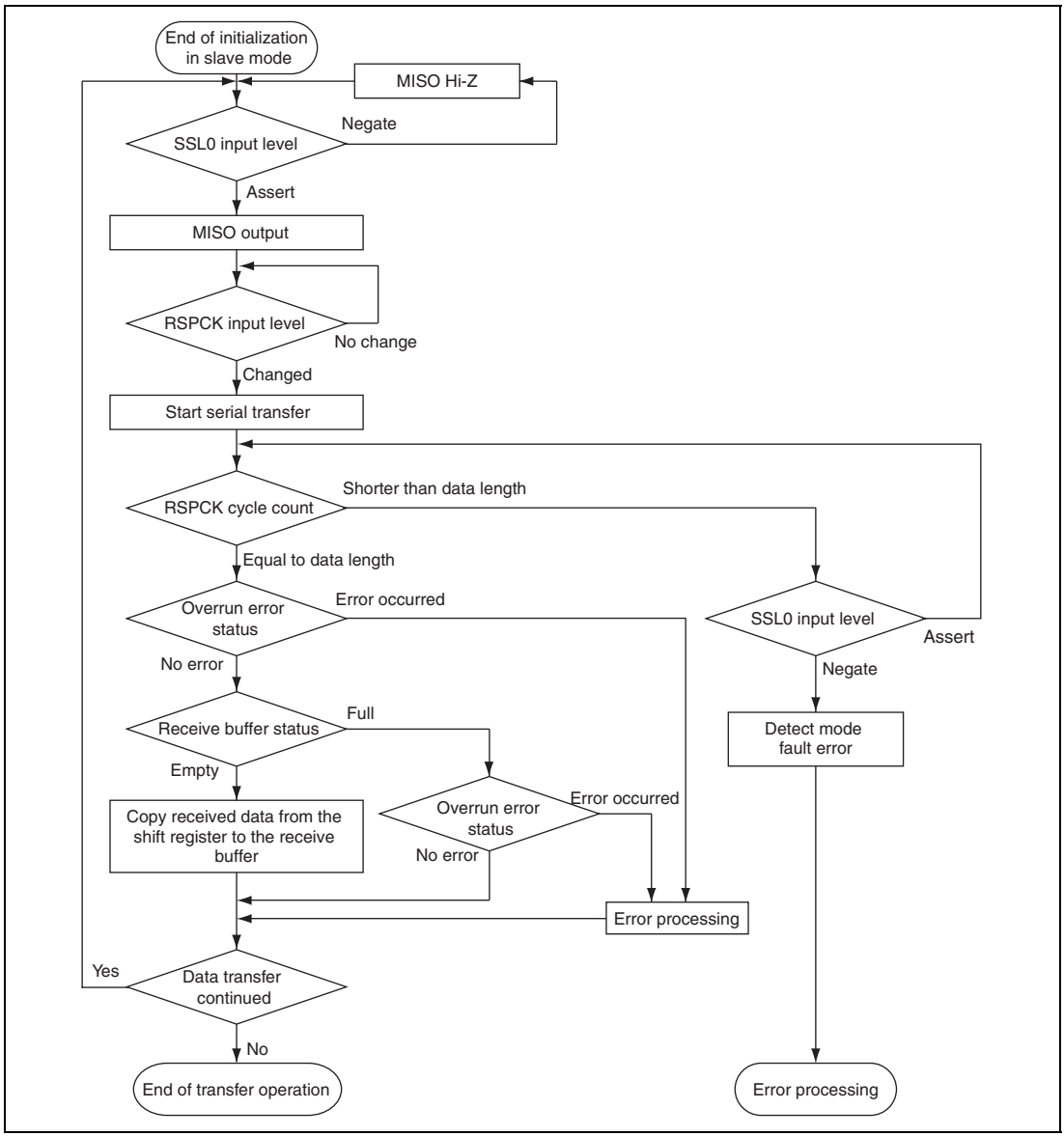


Figure 18.20 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

(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.

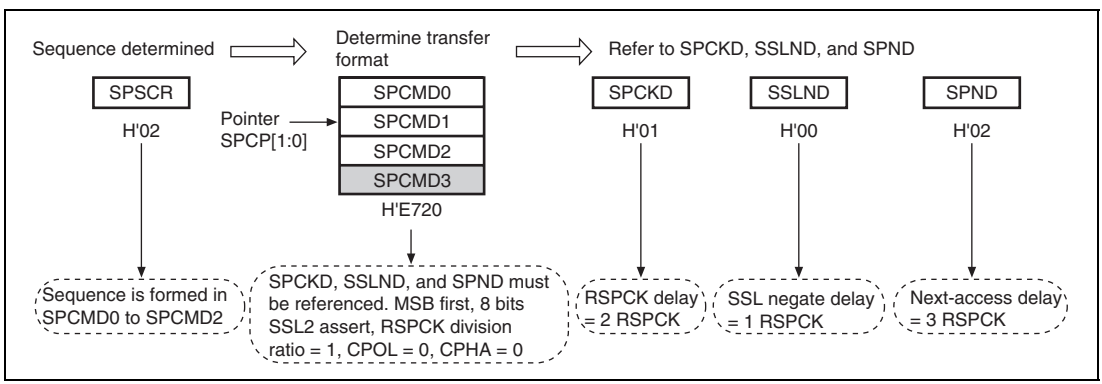


Figure 18.21 Determination Procedure of Serial Transfer Mode in Master Mode

(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.

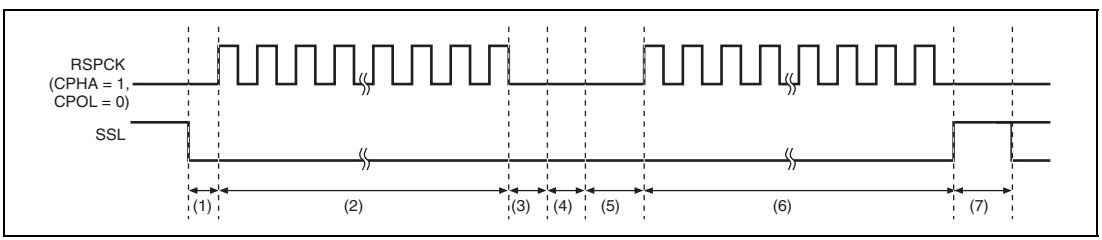


Figure 18.22 Example of Burst Transfer Operation using SSLKP Bit

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.

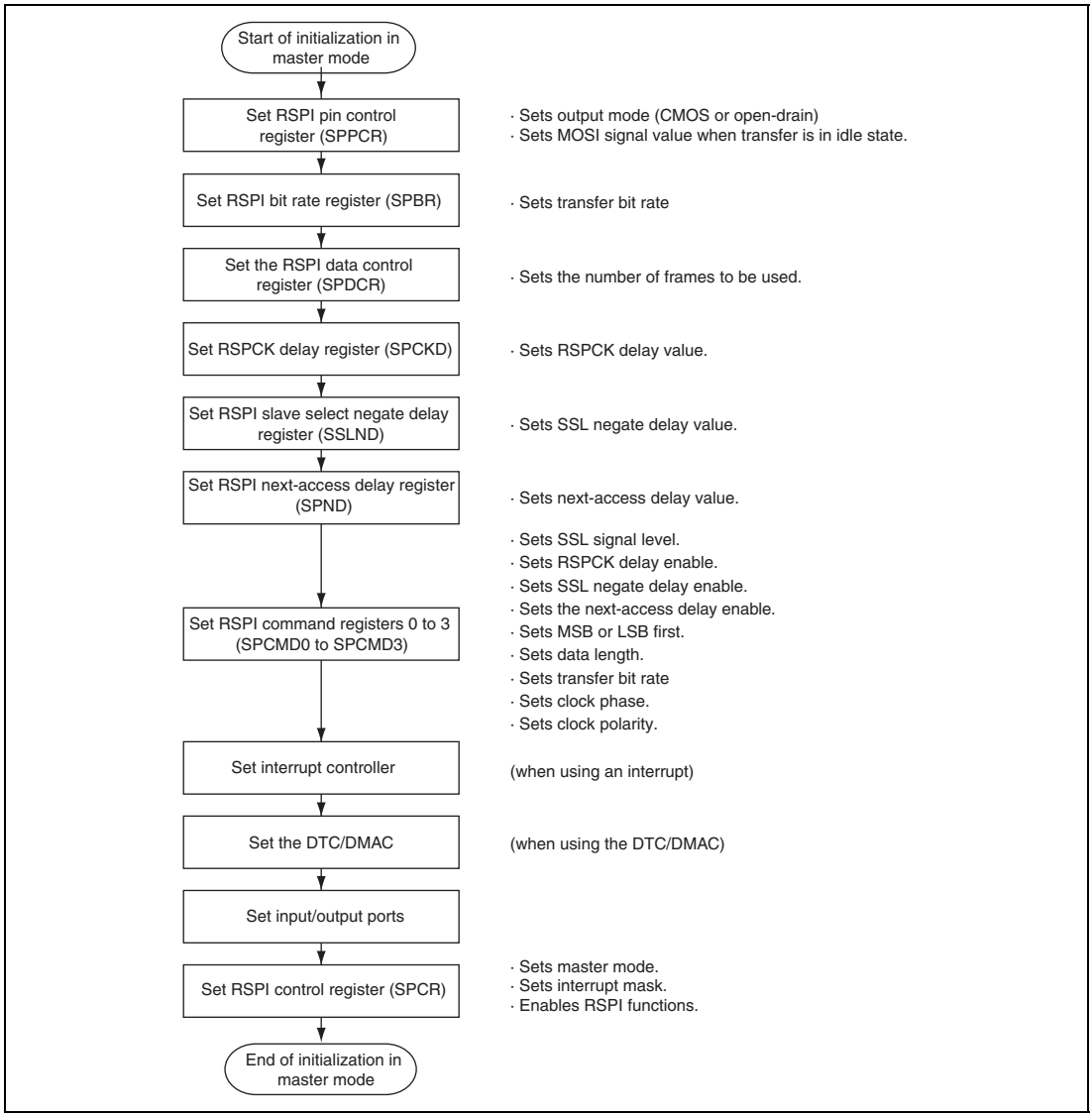
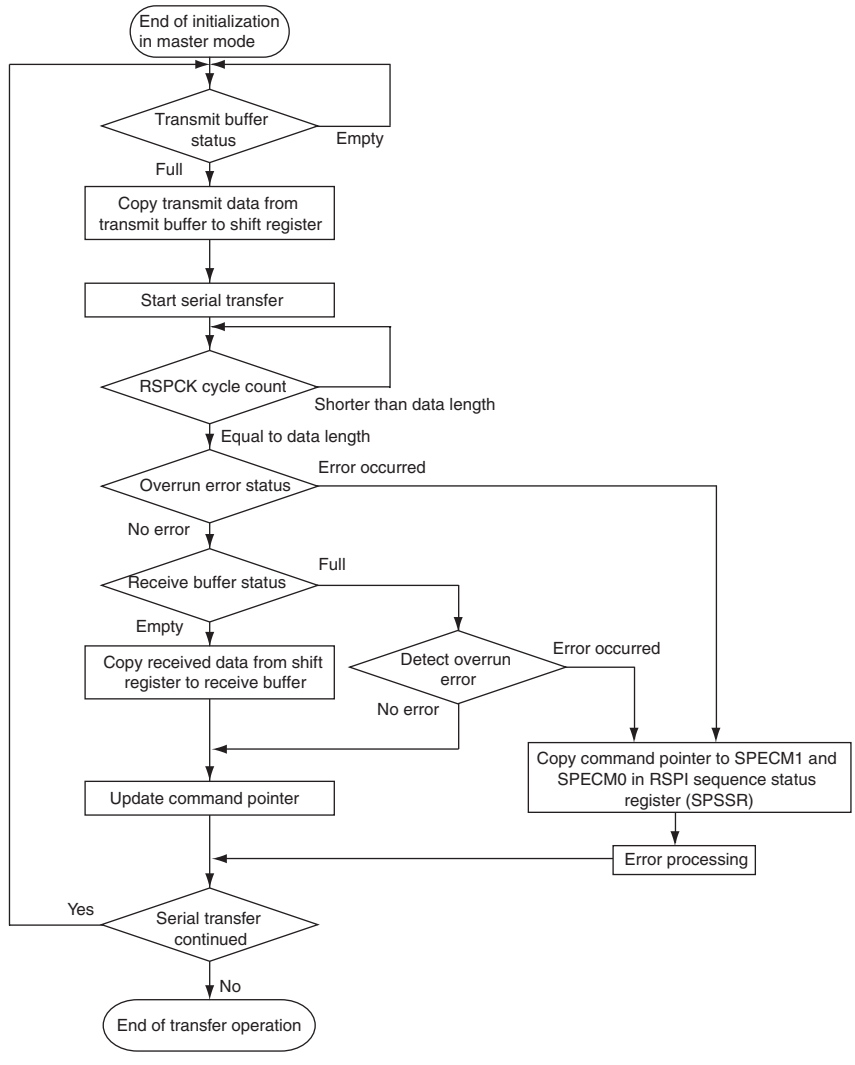


Figure 18.23 Example of Initialization Flowchart in Master Mode

(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.

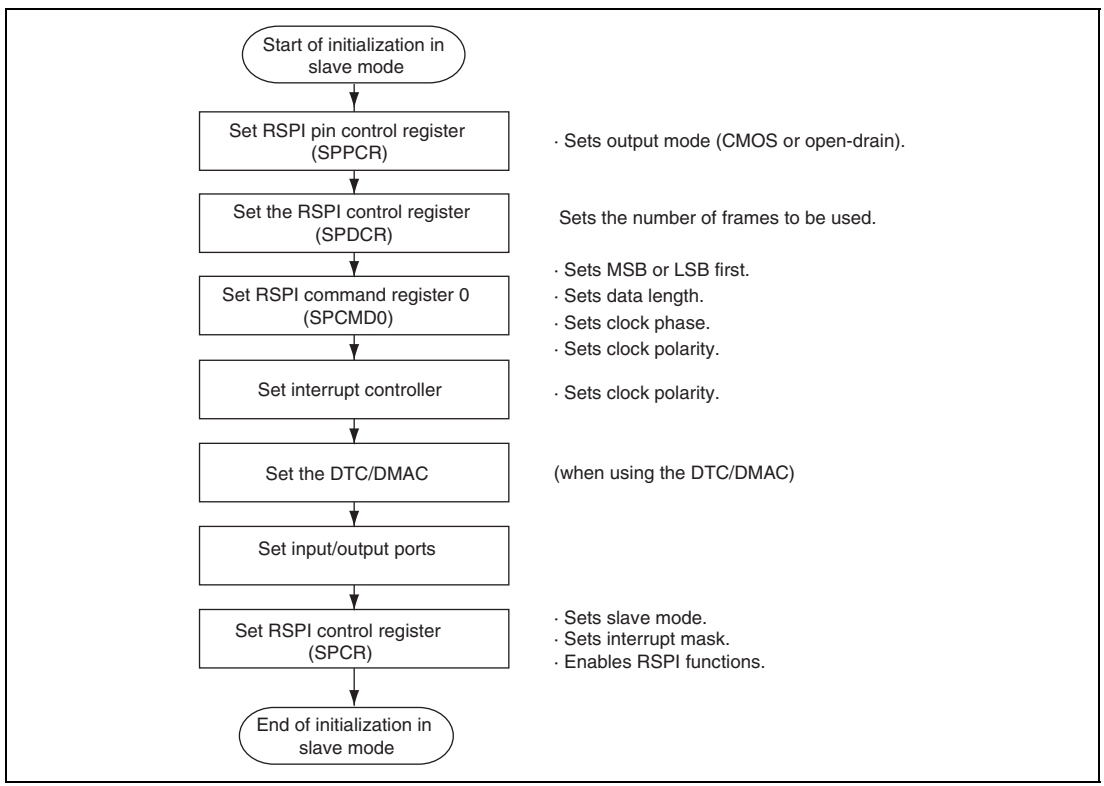


Figure 18.25 Example of Initialization Flowchart in Slave Mode

(1-4) Transfer Operation Flowchart (CPHA = 1)

Figure 18.26 shows an example of transfer operation flowchart for the RSPI during clock synchronous operation.

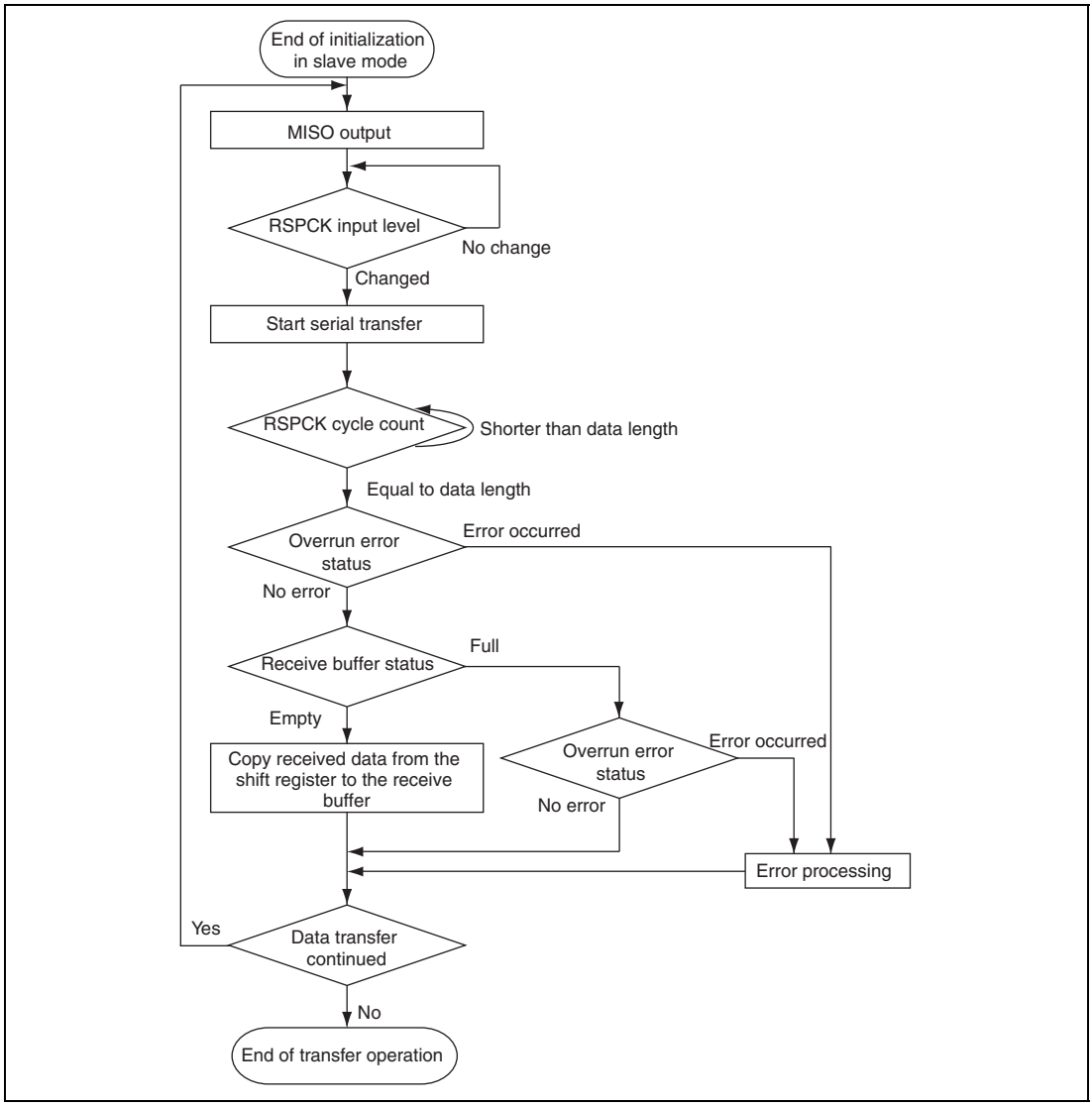


Figure 18.26 Example of Transfer Operation Flowchart in Slave Mode (CPHA = 1)

(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.

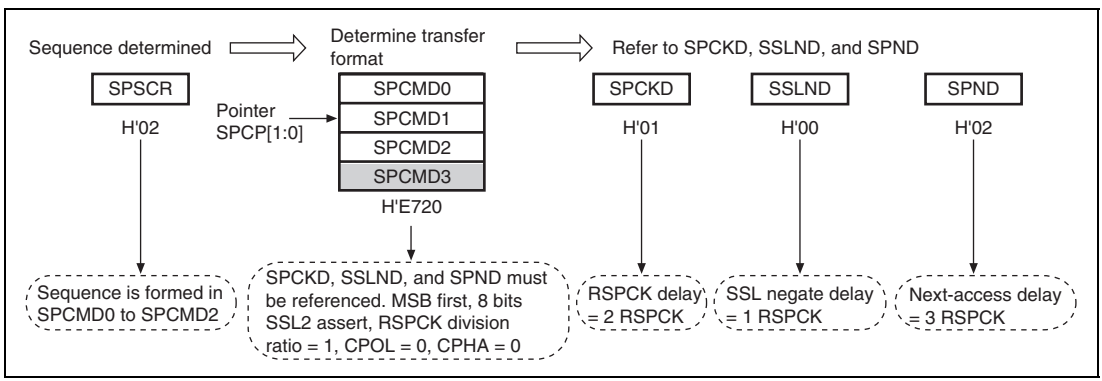


Figure 18.27 Determination Procedure of Serial Transfer Mode in Master Mode

(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.

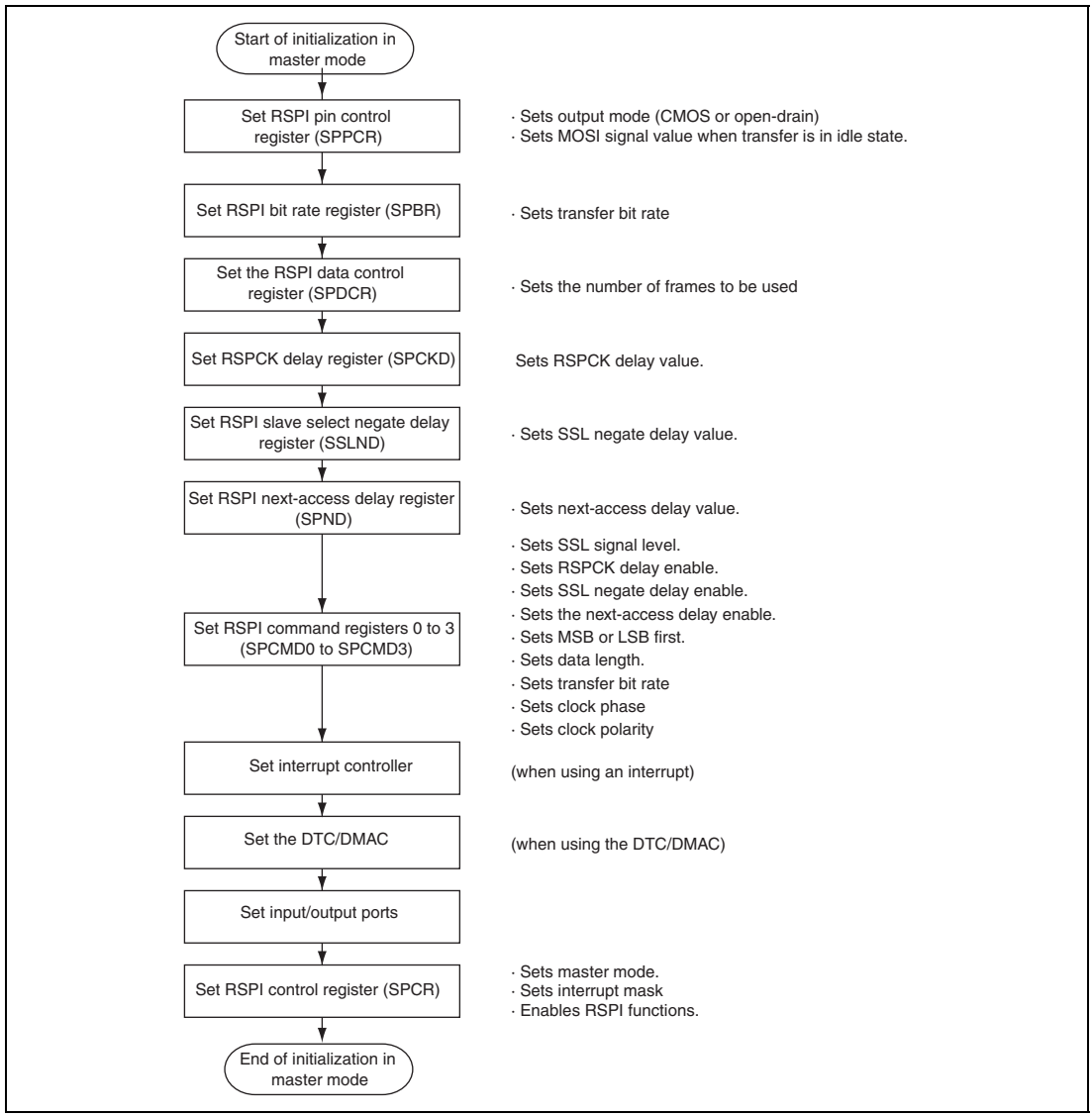


Figure 18.28 Example of Initialization Flowchart in Master Mode

(2-5) Transfer Operation Flowchart

Figure 18.29 shows an example of transfer operation flowchart in master mode during clock synchronous operation.

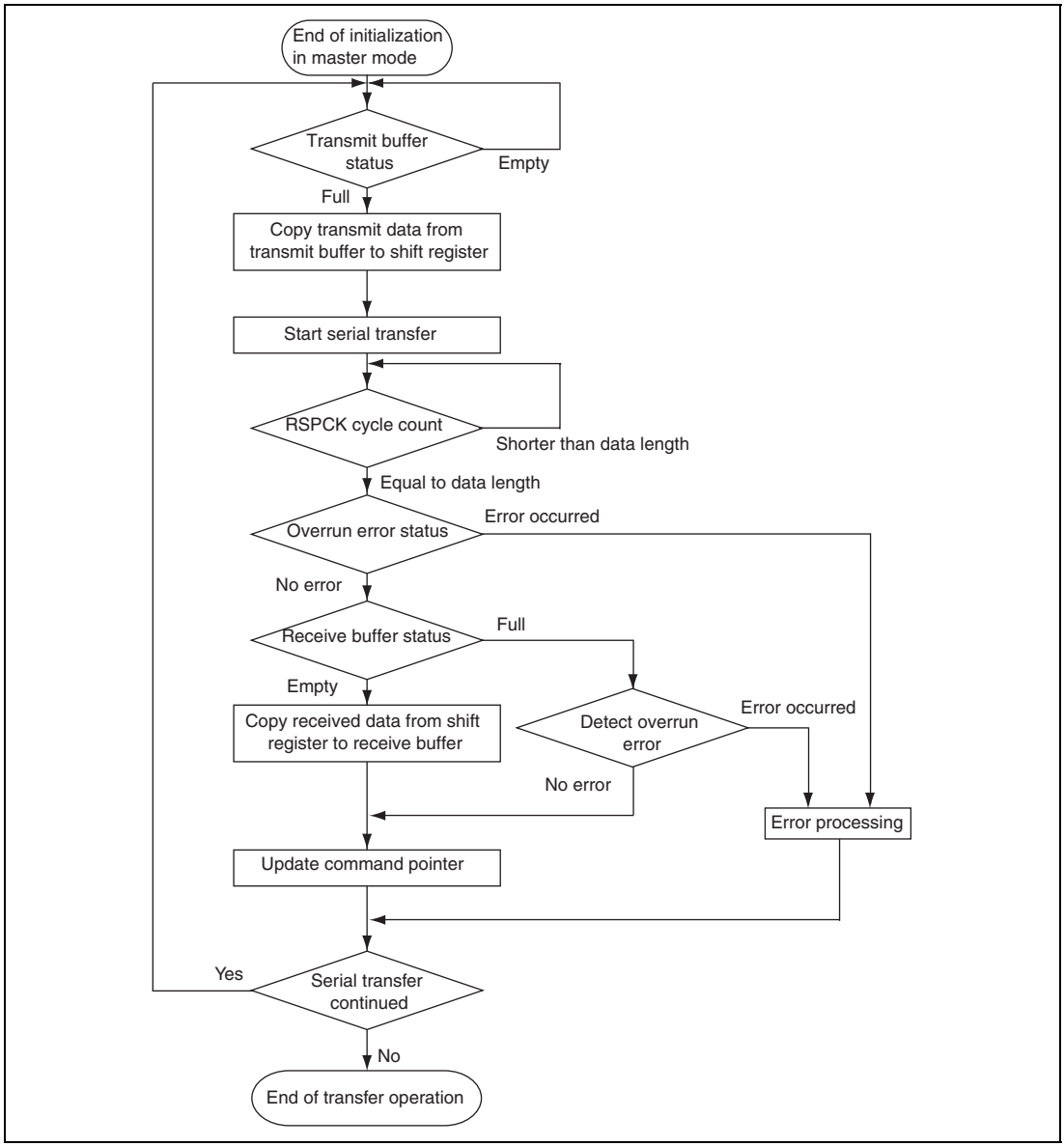


Figure 18.29 Example of Transfer Operation Flowchart in Master Mode

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.

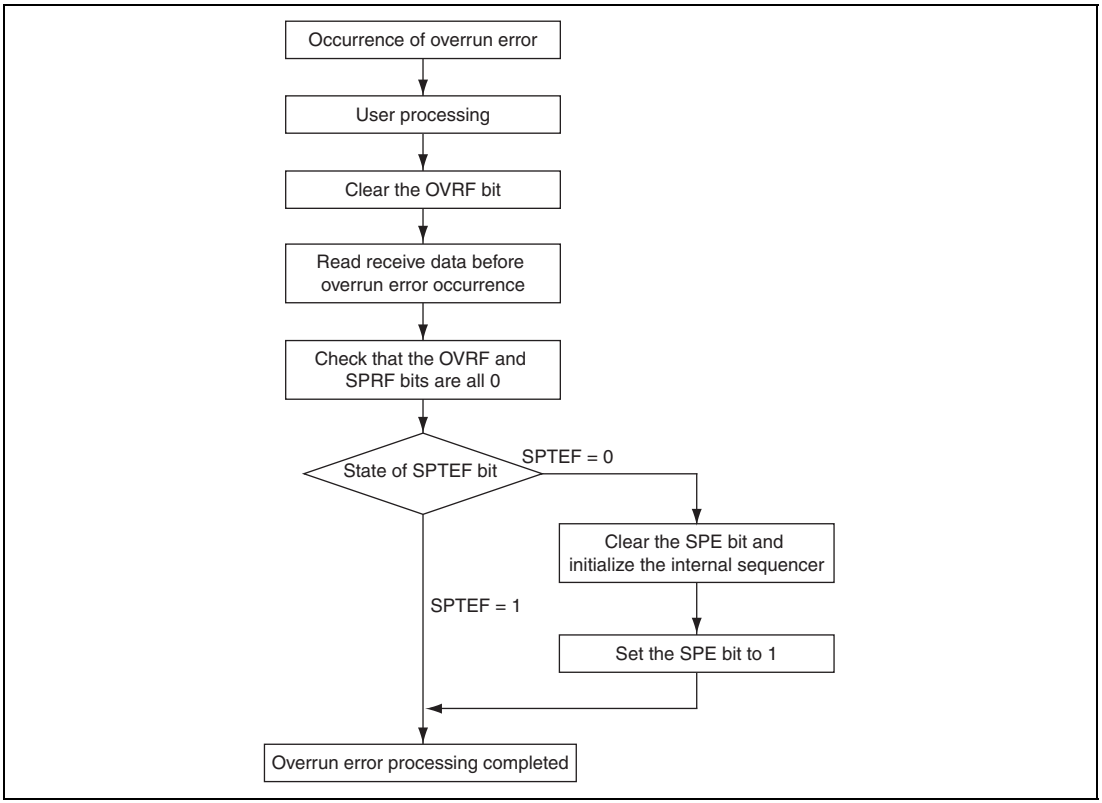


Figure 18.30 Error Processing (Overrun Error)

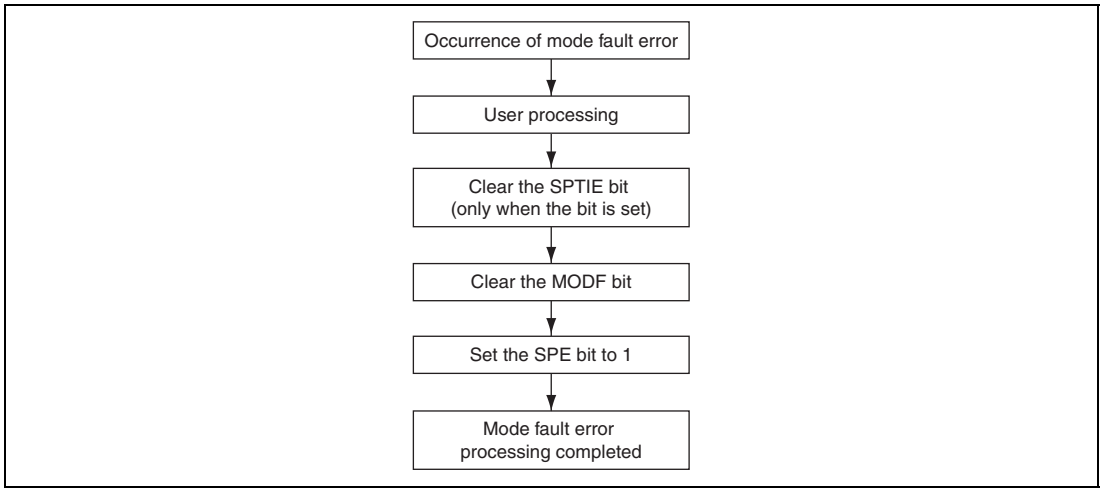


Figure 18.31 Error Processing (Mode Fault Error)

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.

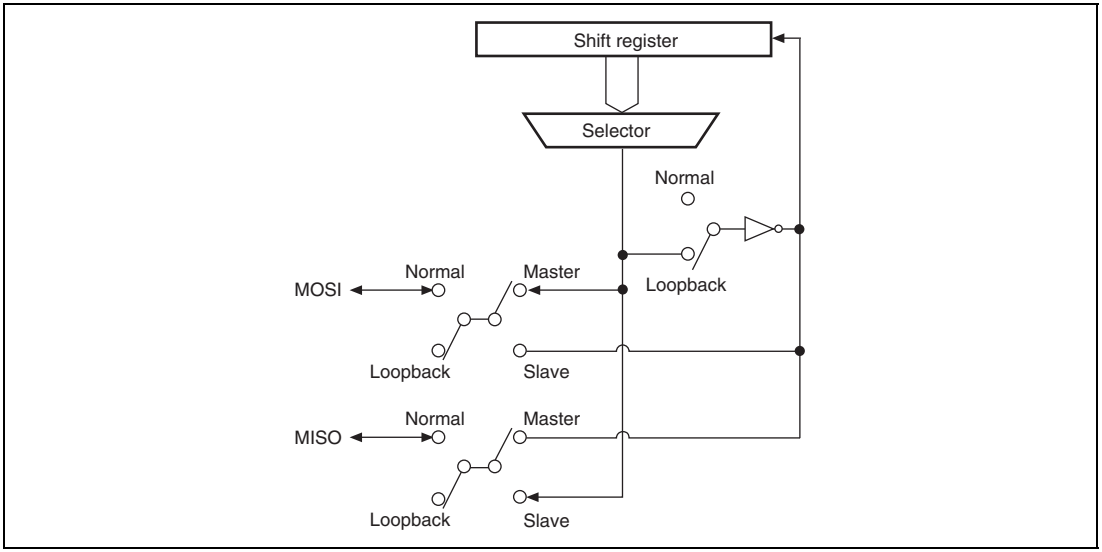


Figure 18.32 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

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.

18.5.6 Unused Pins in Slave Mode

Pins SSL1 to SSL3 are not used when the RSPI is used in slave mode. In that case, use the port E control registers L2 and L3 (PECRL2 and PECRL3) of the pin function controller (PFC) to allocate other pin functions to these pins or to disable them.

Section 19 A/D Converter (ADC)

This LSI includes a successive approximation type 12-bit A/D converter.

19.1 Features

- 12-bit resolution
- Input channels: 16 channels
- High-speed conversion

  When Aϕ = 50 MHz: Minimum 1.0 µs per channel

  AD clock = 50 MHz, 50 conversion states

  When Aϕ = 40 MHz: Minimum 1.25 µs per channel

  AD clock = 40 MHz, 50 conversion states
- Three operating modes
  - 2-channel scan mode: Performs a single A/D conversion on a maximum of two channels
  - Single-cycle scan mode: Performs a single A/D conversion on the specified channel
  - Continuous scan mode: Performs repetitive A/D conversion on the specified channel
- 16 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)
- A/D synchronous conversion

  A/D_2 can be started synchronously with A/D_1.

- 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 19.1 shows a block diagram of the A/D converter.

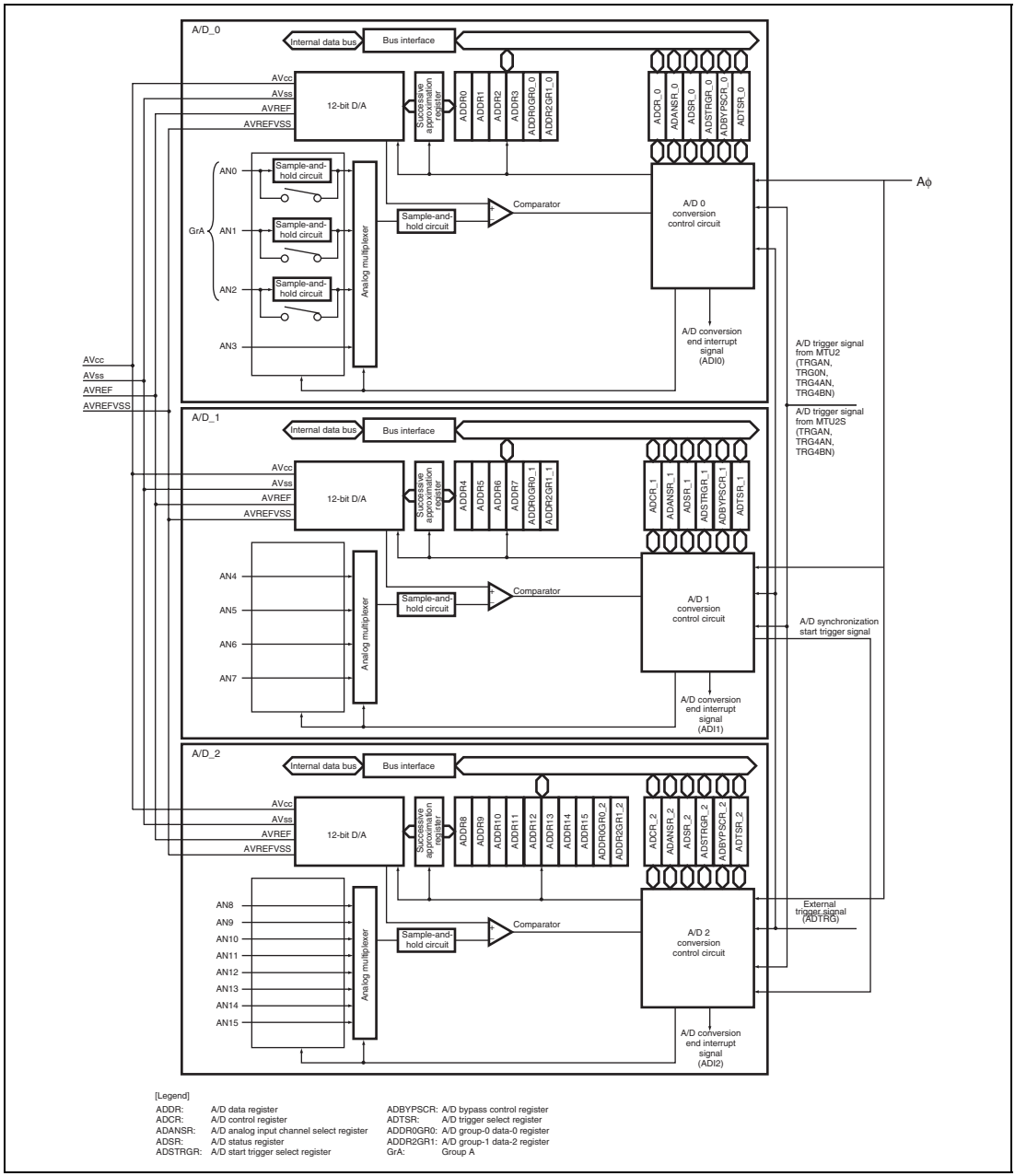


Figure 19.1 Block Diagram of A/D Converter

19.2 Input/Output Pins

Table 19.1 shows the configuration of the pins used by the A/D converter. For the pin usage, refer to the usage notes in section 19.7, Usage Notes.

Table 19.1 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
A/D module 2 (A/D_2)	AN8	Input	Analog input pin 8
	AN9	Input	Analog input pin 9
	AN10	Input	Analog input pin 10
	AN11	Input	Analog input pin 11
	AN12	Input	Analog input pin 12
	AN13	Input	Analog input pin 13
	AN14	Input	Analog input pin 14
	AN15	Input	Analog input pin 15

19.3 Register Descriptions

The A/D converter has the following registers.

Table 19.2 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
A/D control register_2	ADCR_2	R/W	H'00	H'FFFFEE00	8
A/D status register_2	ADSR_2	R/W	H'00	H'FFFFEE02	8
A/D start trigger select register_2	ADSTRGR_2	R/W	H'00	H'FFFFEE1C	8
A/D analog input channel select register_2	ADANSR_2	R/W	H'00	H'FFFFEE20	8
A/D bypass control register_2	ADBYPSCR_2	R/W	H'00	H'FFFFEE30	8
A/D trigger select register_0	ADTSR_0	R/W	H'0000	H'FFFFE930	16
A/D trigger select register_1	ADTSR_1	R/W	H'0000	H'FFFFED30	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D trigger select register_2	ADTSR_2	R/W	H'0000	H'FFFFEF30	16
A/D data register 8	ADDR8	R	H'0000	H'FFFFEE40	16
A/D data register 9	ADDR9	R	H'0000	H'FFFFEE42	16
A/D data register 10	ADDR10	R	H'0000	H'FFFFEE44	16
A/D data register 11	ADDR11	R	H'0000	H'FFFFEE46	16
A/D data register 12	ADDR12	R	H'0000	H'FFFFEE48	16
A/D data register 13	ADDR13	R	H'0000	H'FFFFEE4A	16
A/D data register 14	ADDR14	R	H'0000	H'FFFFEE4C	16
A/D data register 15	ADDR15	R	H'0000	H'FFFFEE4E	16
A/D group-0 data-0 register_0	ADDR0GR0_0	R	H'0000	H'FFFFE932	16
A/D group-0 data-0 register_1	ADDR0GR0_1	R	H'0000	H'FFFFED32	16
A/D group-0 data-0 register_2	ADDR0GR0_2	R	H'0000	H'FFFFEF32	16
A/D group-1 data-2 register_0	ADDR2GR1_0	R	H'0000	H'FFFFE934	16
A/D group-1 data-2 register_1	ADDR2GR1_1	R	H'0000	H'FFFFED34	16
A/D group-1 data-2 register_2	ADDR2GR1_2	R	H'0000	H'FFFFEF34	16

19.3.1 A/D Control Registers 0 to 2 (ADCR_0 to ADCR_2)

ADCR_0 to ADCR_2 are 8-bit readable/writable registers that select 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	R/W	R/W	R/W	R	R	R/W	R/W

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 and 2-channel 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.
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.
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.

Bit	Bit Name	Initial Value	R/W	Description
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.
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

Bit	Bit Name	Initial Value	R/W	Description
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. This bit setting is invalid in 2-channel scan mode. 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 ($\overline{\text{ADTRG}}$)

19.3.2 A/D Status Registers 0 to 2 (ADSR_0 to ADSR_2)

ADSR_0 to ADSR_2 are 8-bit readable/writable registers that indicate 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

19.3.3 A/D Start Trigger Select Registers 0 to 2 (ADSTRGR_0 to ADSTRGR_2)

ADSTRGR_0 to ADSTRGR_2 select 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, the 2CHSE bit in ADTSR is set to 0, 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).

19.3.4 A/D Analog Input Channel Select Registers 0 to 2 (ADANSR_0 to ADANSR_2)

ADANSR_0 to ADANSR_2 are 8-bit readable/writable registers that select an analog input channel.

Bit:	7	6	5	4	3	2	1	0
	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
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	ANS7	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 19.3. When changing the analog input channel, the ADST bit in ADCR must be cleared to 0 to prevent incorrect operations. In ADANSR_0 and ADANSR_1, bits 7 to 4 (ANS7 to ANS4) are reserved. These bits are always read as 0 and the write value should always be 0.
6	ANS6	0	R/W	
5	ANS5	0	R/W	
4	ANS4	0	R/W	
3	ANS3	0	R/W	
2	ANS2	0	R/W	
1	ANS1	0	R/W	
0	ANS0	0	R/W	

Table 19.3 Channel Select List

	Analog Input Channels		
Bit Name	**A/D_0**	**A/D_1**	**A/D_2**
ANS0	AN0	AN4	AN8
ANS1	AN1	AN5	AN9
ANS2	AN2	AN6	AN10
ANS3	AN3	AN7	AN11
ANS4	—	—	AN12
ANS5	—	—	AN13
ANS6	—	—	AN14
ANS7	—	—	AN15

19.3.5 A/D Bypass Control Registers 0 to 2 (ADBYPSCR_0 to ADBYPSCR_2)

ADBYPSCR_0 to ADBYPSCR_2 determine whether to use the sample-and-hold circuits and A/D conversion synchronization function for the corresponding channels.

For A/D conversion of group A (GrA), it can be selected whether 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 19.4, Operation.

Setting the ADSST bit to 1 in ADBYPSCR_2 allows the conversion start timing of A/D converter 2 to be synchronized with that of A/D converter 1. For details, refer to section 19.4, Operation.

The function of the SH bit in this register is available only for A/D converter_0. A/D converter_1 and A/D converter_2 are always in the same state as when the SH bit is set to 0.

The function of the ADSST bit in this register is available only for A/D converter_2. A/D converter_0 and A/D converter_1 are always in the same state as when the ADSST bit is set to 0.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	ADSST	-	-	SH
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	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	ADSST	0	R/W	A/D Synchronization Start (only in ADBYPSCR_2) Allows A/D_2 to start synchronously with A/D_1. This bit should be set when the ADST bit in ADCR_2 is 0. When A/D synchronous conversion starts, this bit is cleared to 0 setting the ADST bit to 1. To prevent A/D conversion from starting by any other factor, the TRGE bit in ADCR_2 should be cleared 0. This bit is reserved in ADBYPSCR_0 and ADBYPSCR_1. The write value should always be 0.
2, 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 (only in ADBYPSCR_0) 0: Does not select the sample-and-hold circuit. 1: Selects the sample-and-hold circuit. This bit is reserved in ADBYPSCR_1 and ADBYPSCR_2. The write value should always be 0.

19.3.6 A/D Data Registers 0 to 15 (ADDR0 to ADDR7)

ADDR0 to ADDR15 are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. (See table 19.4.)

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 19.4 Correspondence between Analog Channels and Registers (ADDR0 to ADDR15)

Analog Input Channels	A/D Data Registers
AN0	ADDR0/ADDR0GR0_0*
AN1	ADDR1
AN2	ADDR2/ADDR2GR1_0*
AN3	ADDR3
AN4	ADDR4/ADDR0GR0_1*
AN5	ADDR5
AN6	ADDR6/ADDR2GR1_1*
AN7	ADDR7
AN8	ADDR8/ADDR0GR0_2*
AN9	ADDR9
AN10	ADDR10/ADDR2GR1_2*
AN11	ADDR11
AN12	ADDR12
AN13	ADDR13
AN14	ADDR14
AN15	ADDR15

Note: * A/D conversion result in 2-channel scan mode can be stored.

19.3.7 A/D Trigger Select Registers 0 to 2 (ADTSR_0 to ADTSR_2)

ADTSR_0 to ADTSR_2 are 16-bit readable/writable registers that enable 2-channel scan mode and set various parameters for 2-channel scan mode.

Note: A/D trigger intervals are equal to or more than (A/D conversion time × N + 30 × M + 14) Aϕ states
When SH in ADBYPSCR = 0, M = 0. When SH in ADBYPSCR = 1, M = 1.
When CHSEC in ADTSR = 0, N = 1. When CHSEC in ADTSR = 1, N = 2.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG1S[3:0]				TRG0S[3:0]				-	-	-	-	-	CHSEC	CON ADF	2CHSE
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/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	TRG1S [3:0]	0000	R/W	A/D Trigger Group 1 Select Select an A/D conversion start trigger from the external, MTU2, and MTU2S triggers for group 1 when the A/D module is in 2-channel scan mode. 0000: Selects the external trigger ($\overline{\text{ADTRG}}$) input. 0001: Enables starting of A/D conversion by the TRGAN trigger (MTU2). 0010: Enables starting of A/D conversion by the TRG0N trigger (MTU2). 0011: Enables starting of A/D conversion by the TRG4AN trigger (MTU2). 0100: Enables starting of A/D conversion by the TRG4BN trigger (MTU2). 0101: Enables starting of A/D conversion by the TRG4AN trigger (MTU2) and TRG4BN trigger (MTU2). 0110: Enables starting of A/D conversion by the TRGAN trigger (MTU2S). 0111: Enables starting of A/D conversion by the TRG4AN trigger (MTU2S). 1000: Enables starting of A/D conversion by the TRG4BN trigger (MTU2S). 1001: Enables starting of A/D conversion by the TRG4AN trigger (MTU2S) and TRG4BN trigger (MTU2S). Other than above: Disables starting of A/D conversion by the external, MTU2, or MTU2S trigger.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	TRG0S [3:0]	0000	R/W	A/D Trigger Group 0 Select Select an A/D conversion start trigger from the external, MTU2, and MTU2S triggers for group 0 when the A/D module is in 2-channel scan mode. 0000: Selects the external trigger ($\overline{\text{ADTRG}}$) input. 0001: Enables starting of A/D conversion by the TRGAN trigger (MTU2). 0010: Enables starting of A/D conversion by the TRG0N trigger (MTU2). 0011: Enables starting of A/D conversion by the TRG4AN trigger (MTU2). 0100: Enables starting of A/D conversion by the TRG4BN trigger (MTU2). 0101: Enables starting of A/D conversion by the TRG4AN trigger (MTU2) and TRG4BN trigger (MTU2). 0110: Enables starting of A/D conversion by the TRGAN trigger (MTU2S). 0111: Enables starting of A/D conversion by the TRG4AN trigger (MTU2S). 1000: Enables starting of A/D conversion by the TRG4BN trigger (MTU2S). 1001: Enables starting of A/D conversion by the TRG4AN trigger (MTU2S) and TRG4BN trigger (MTU2S). Other than above: Disables starting of A/D conversion by the external, MTU2, or MTU2S trigger.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	CHSEC	0	R/W	Channel Select in 2-Channel Scan Mode Selects the channels for group 0 and group 1 in 2-channel scan mode. See table 19.5.

Bit	Bit Name	Initial Value	R/W	Description
1	CONADF	0	R/W	ADF Control Controls the ADF operation in 2-channel scan mode. This bit is valid only when 2-channel scan mode is selected and starting of A/D conversion by a trigger is enabled (TRGE = 1). 0: ADF is set when either of the conversion by group 0 trigger or conversion by group 1 trigger is completed. 1: ADF is set when both of the conversion by group 0 trigger and conversion by group 1 trigger are completed.
0	2CHSE	0	R/W	2-Channel Scan Mode Enable Enables the 2-channel scan mode function. Single-cycle scan mode and continuous scan mode are ignored. 0: The ADC operates in 1-channel scan mode or continuous scan mode according to the ADCS bit setting in ADCR. 1: The ADC operates in 2-channel scan mode. The ADCS bit setting in ADCR is ignored.

Table 19.5 Selecting Channels in 2-Channel Scan Mode

	Analog Input Channel		
	2-Channel Scan Mode (2CHSE = 1), Started by Trigger		
CHSEC bit	A/D_0	A/D_1	A/D_2
0	Group 0: AN0 Group 1: AN2	Group 0: AN4 Group 1: AN6	Group 0: AN8 Group 1: AN10
1	Group 0: AN0, AN1 Group 1: AN2, AN3	Group 0: AN4, AN5 Group 1: AN6, AN7	Group 0: AN8, AN9 Group 1: AN10, AN11

Note: * The ADCS bit setting is ignored and single-cycle scan mode is selected.

	Analog Input Channel		
	2-Channel Scan Mode (2CHSE = 1), Started by Software, Scanning Operation on Channels Corresponding to Bits ANS0 to ANS7 in ADANSR		
Bit Name	**A/D_0**	**A/D_1**	**A/D_2**
ANS0	AN0	AN4	AN8
ANS1	AN1	AN5	AN9
ANS2	AN2	AN6	AN10
ANS3	AN3	AN7	AN11
ANS4	—	—	AN12
ANS5	—	—	AN13
ANS6	—	—	AN14
ANS7	—	—	AN15

Notes: 1. The ADF bit is set when CONADF is 0.
The ADF bit is not set when CONADF is 1.

2. When 2-channel scan mode is selected and conversion is then started by software, A/D conversion is not performed for the individual groups (group 0 and group 1). Single-cycle scan mode is used instead when A/D conversion of signals on pins for channels corresponding to the setting of the ANS bits is required.
The trigger selected according to the ADTSR settings only enables activation of A/D conversion by group 0 or group 1.

19.3.8 A/D Group-0 Data-0 Registers 0 to 2 (ADDR0GR0_0 to ADDR0GR0_2)

ADDR0GR0_0 (ADDR0GR0_1, ADDR0GR0_2) is a 16-bit readable register. In 2-channel scan mode, the conversion result of AN0 (AN4, AN8) is stored in ADDR0GR0_0 (ADDR0GR0_1, ADDR0GR0_2) when the TRG0S bits in ADTSR are set to B'0101 and starting of A/D conversion by TRG4BN of the MTU2 is enabled, or when the TRG0S bits in ADTSR are set to B'1001 and starting of A/D conversion by TRG4BN of the MTU2S is enabled. 12-bit conversion data is stored in bits 11 to 0 in ADDR0GR0_0 (ADDR0GR0_1, ADDR0GR0_2).

When the ADE bit in ADCR is 1, this register can be automatically cleared to H'0000 after it is read.

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

19.3.9 A/D Group-1 Data-2 Registers 0 to 2 (ADDR2GR1_0 to ADDR2GR1_2)

ADDR2GR1_0 (ADDR2GR1_1, ADDR2GR1_2) is a 16-bit readable register. In 2-channel scan mode, the conversion result of AN2 (AN6, AN10) is stored in ADDR2GR1_0 (ADDR2GR1_1, ADDR2GR1_2) when the TRG1S bits in ADTSR are set to B'0101 and starting of A/D conversion by TRG4BN of the MTU2 is enabled, or when the TRG1S bits in ADTSR are set to B'1001 and starting of A/D conversion by TRG4BN of the MTU2S is enabled.

When the ADE bit in ADCR is 1, this register can be automatically cleared to H'0000 after it is read.

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

19.4 Operation

The A/D converter has three operating modes: single-cycle scan mode, continuous scan mode, and 2-channel 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. In 2-channel scan mode, four channels of analog inputs are divided into two groups, group 0 and group 1, and a single A/D conversion is performed on the channels selected by the triggers that have been set separately for group 0 and group 1.

The operating mode is selected by the ADCS bit in the A/D control register (ADCR) and the 2CHSE bit in the A/D trigger select register (ADTSR). Setting the ADCS and 2CHSE bits to 0 selects single-cycle scan mode, setting the ADCS bit to 1 and the 2CHSE bit to 0 selects continuous scan mode, and setting the 2CHSE bit to 1 with any ADCS bit setting selects 2-channel scan mode. In both single-cycle scan mode and continuous scan mode, A/D conversion starts on the channel with the smallest number in the analog input channels selected by the A/D analog input channel select register (ADANSR) (AN0 to AN15). When the ADST bit in ADCR is set to 1 by the MTU2, MTU2S, or external trigger input, A/D conversion starts on the channel with the smallest number in the analog input channels selected by the CHSEC bit in the A/D trigger select register (ADTSR) (for example, AN0, AN1). When the ADST bit in ADCR is set to 1 by software, A/D conversion starts on the channel with the smallest number in the analog input channels selected by the A/D analog input channel select register (ADANSR) from AN0 to AN15.

A/D_2 can be started synchronously with A/D_1 by setting the ADSST bit to 1 in A/D bypass control register 2 (ADBYPSCR_2). In this case, it is necessary for A/D_1 to perform A/D conversion at least once. A/D_2 starts conversion synchronously with A/D_1 at the A/D synchronization start trigger signal of A/D_1 that is generated after setting the ADSST bit.

The operating mode should be set before setting the ADSST bit. The ADSST bit should be set when the ADST bit is 0 in the A/D control register (ADCR_2). When A/D synchronous conversion starts, the ADSST bit is cleared to 0 setting the ADST bit to 1. To prevent A/D conversion from starting by any factor other than the A/D synchronization start trigger signal, the TRGE bit should be cleared 0 in ADCR_2.

In single-cycle scan mode and 2-channel 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.

19.4.1 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 19.2 and 19.3.

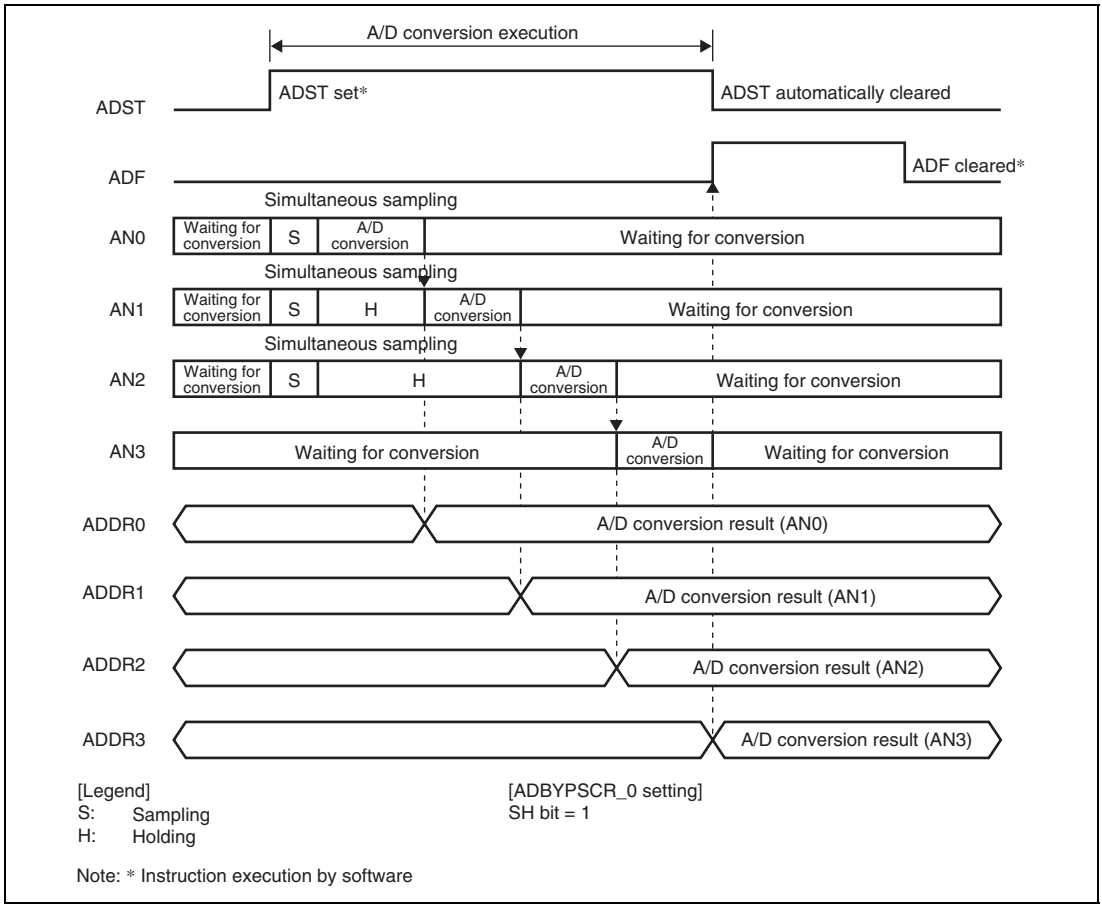


Figure 19.2 Example 1 of A/D_0 Converter Operation (Single-Cycle Scan Mode and Sample-and-Hold Circuit Enabled)

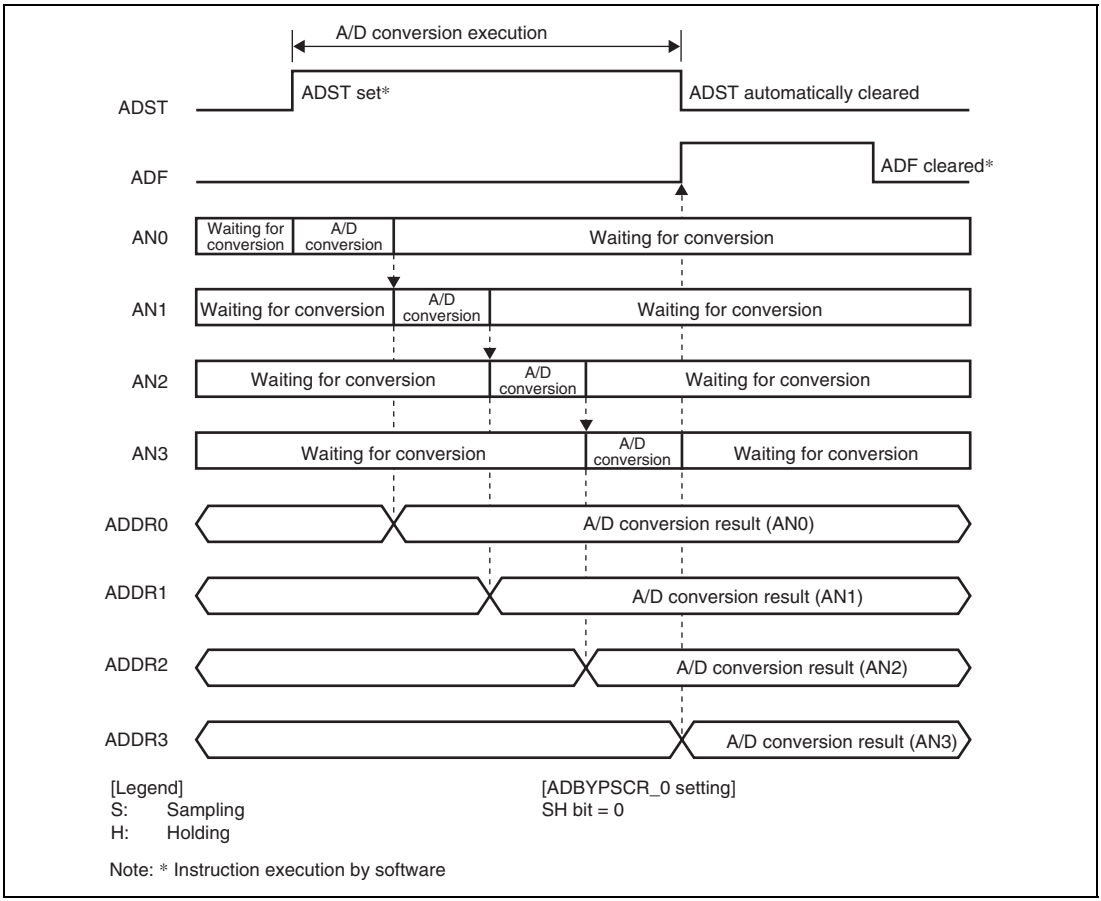


Figure 19.3 Example 2 of A/D_0 Converter Operation (Single-Cycle Scan Mode and Sample-and-Hold Circuit Disabled)

19.4.2 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 19.4 and 19.5.

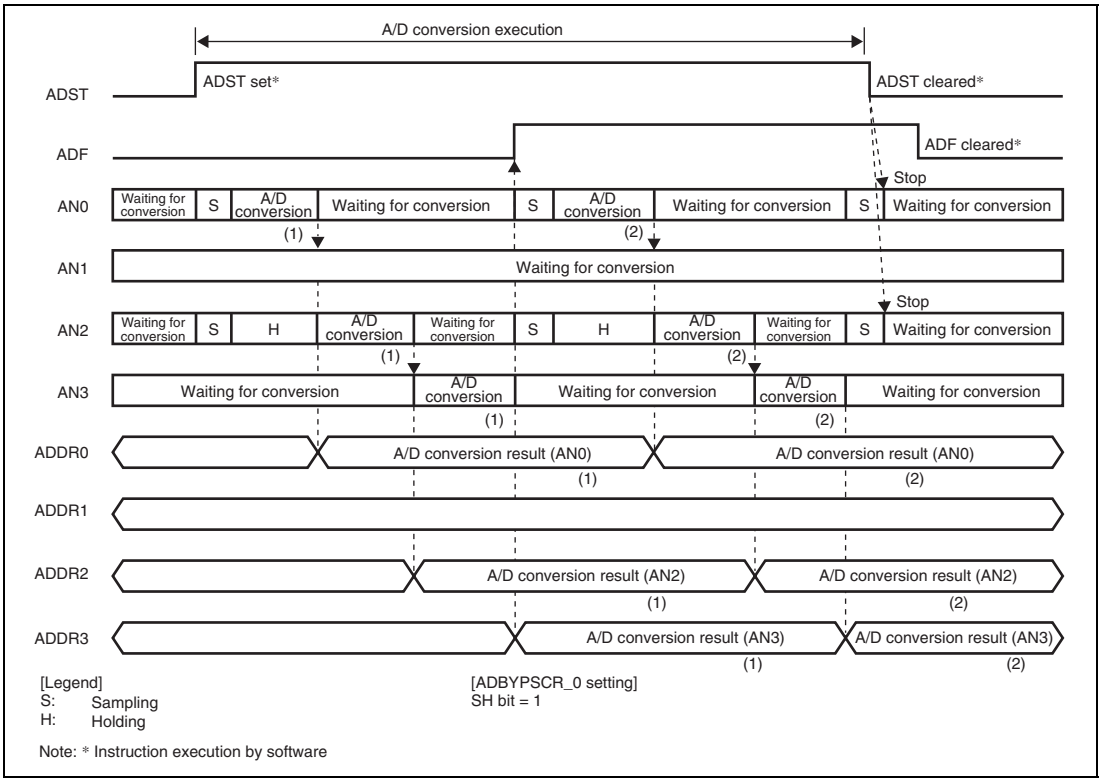


Figure 19.4 Example 1 of A/D_0 Converter Operation (Continuous Scan Mode and Sample-and-Hold Circuit Enabled)

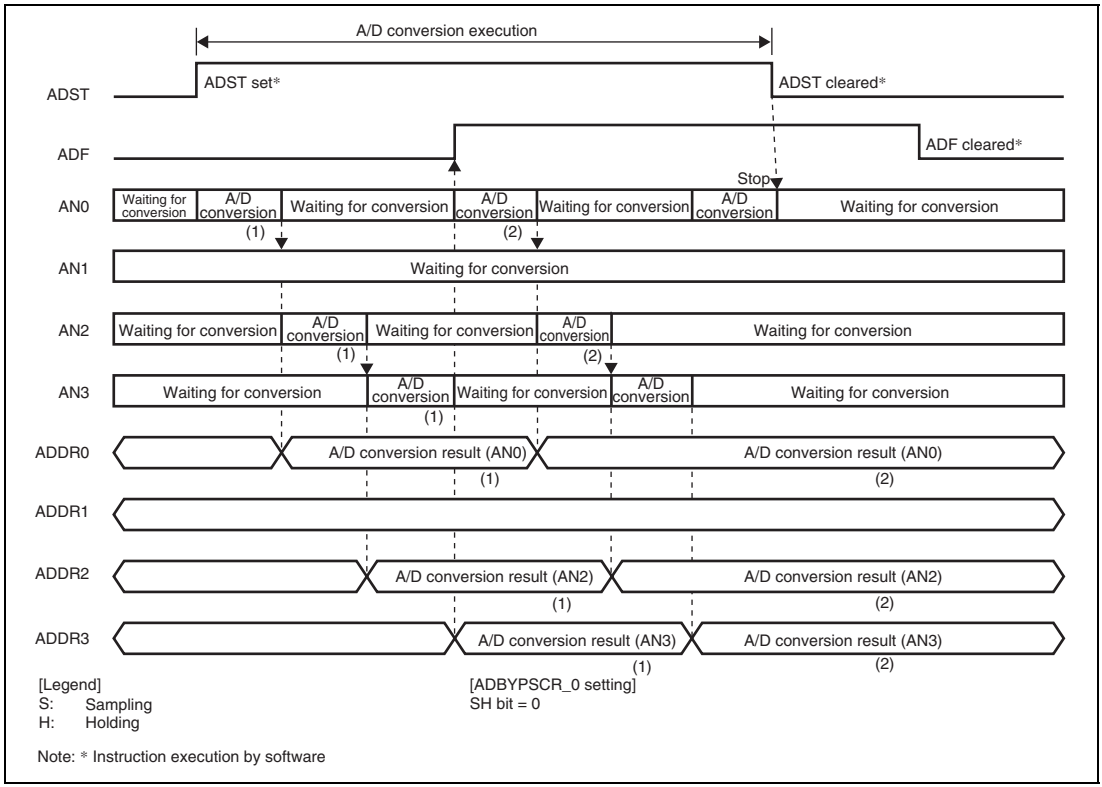


Figure 19.5 Example 2 of A/D_0 Converter Operation (Continuous Scan Mode and Sample-and-Hold Circuit Disabled)

19.4.3 2-Channel Scan Mode

(1) Operation Example in 2-Channel Scan Mode

In 2-channel scan mode, four channels of analog inputs are divided into two groups, group 0 and group 1. The activation source (trigger) can be separately selected for group 0 and group 1. Two-channel scan mode conversion end interrupt can be generated on completion of group-0 or group-1 conversion or on completion of both group-0 and group-1 conversions. To start conversion by a trigger, different sources should be set for group 0 and group 1 in ADTSR. If a group-1 conversion request occurs during group-0 conversion, the group-1 conversion request is ignored. Figure 19.6 shows an operation example when the A/D conversion start request for group 0 is TRG4AN of the MTU2 and A/D conversion start request for group 1 is TRG4BN of the MTU2.

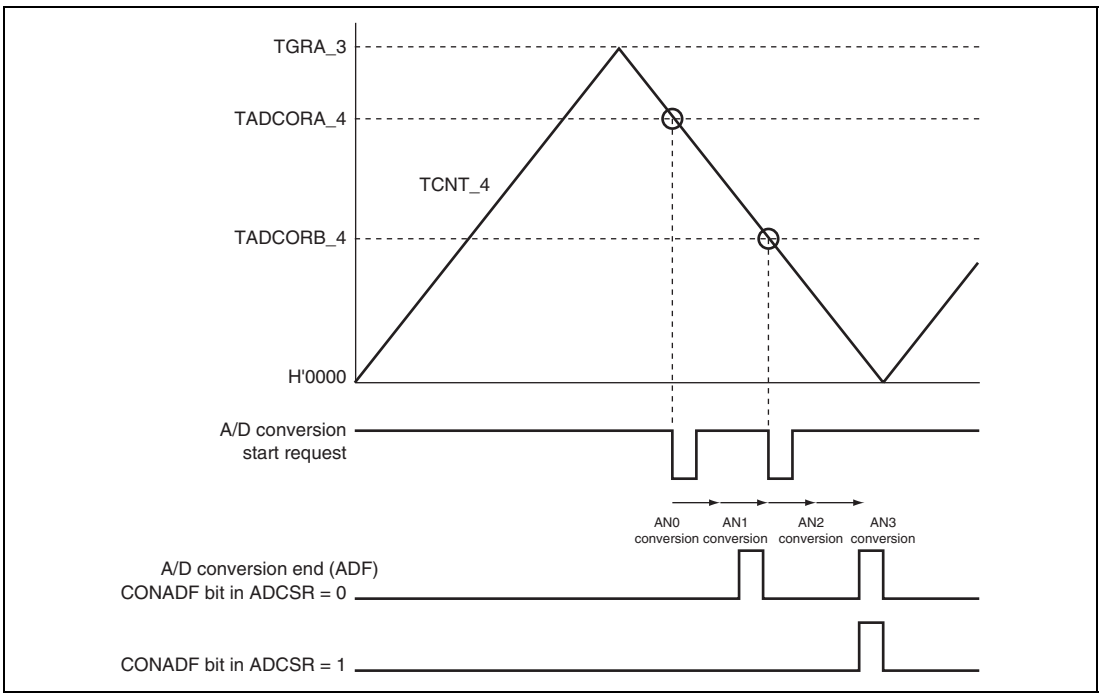


Figure 19.6 2-Channel Scan Mode Operation Example (1)

When the TRG0S bits are set to B'0101 for group 0 to select TRG4AN and TRG4BN as the triggers and TRG1S bits are set for group 1 to select TRGAN, three interrupts are generated. However, when CONADF = 1, the ADI interrupt is generated only when both group-0 conversion and group-1 conversion are completed.

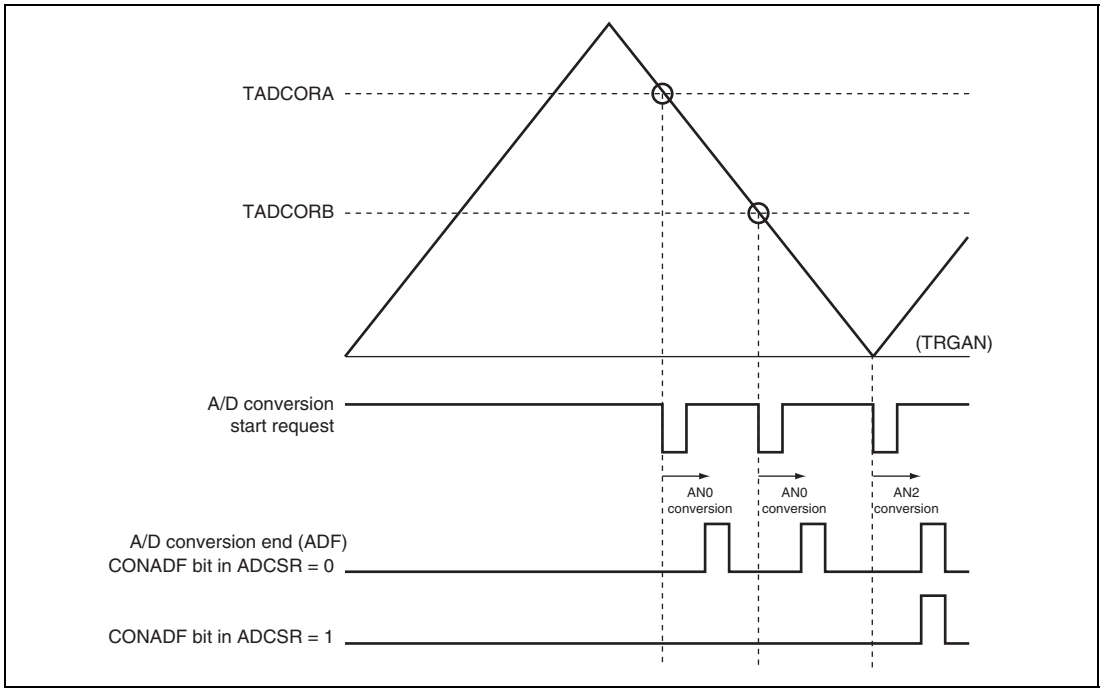


Figure 19.7 2-Channel Scan Mode Operation Example (2)

(2) ADDR0GR0_0 and ADDR2GR1_0 Operation and ADF Setting

Figure 19.8 shows correspondences between operation of ADDR0GR0_0 and ADDR2GR1_0 and setting of ADF flag.

In 2-channel scan mode, when the TRG1S bits or TRG0S bits are B'0101 in ADTSR, starting of A/D conversion by two triggers (TRG4AN and TRG4BN triggers) is enabled. The results of A/D conversion started by these triggers are stored in ADDR0 and ADDR0GR0_0. The registers can be read after these two A/D conversions are completed. The corresponding ADF flag is set after each conversion ends. When ADIE = 1, the A/D conversion end interrupt (ADI) is generated after the conversion ends. The ADI interrupt can be cleared by clearing the ADF flag or ADIE bit to 0.

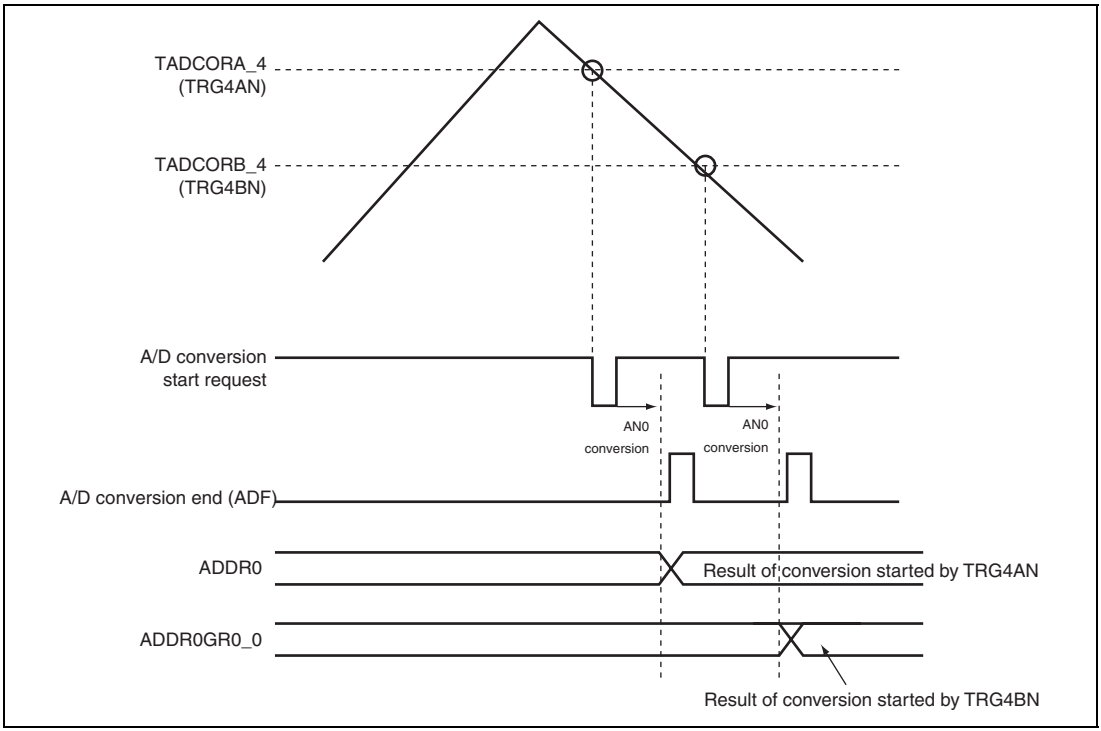


Figure 19.8 ADDR0GR0_0 and ADDR2GR1_0 Operation and ADF Flag Setting

(3) Notes on Selection of 2-Channel Scan Mode

When 2-channel scan mode is selected and conversion is then started by software, A/D conversion is not performed for the individual groups (group 0 and group 1). Single-cycle scan mode is used instead when A/D conversion of signals on pins for channels corresponding to the setting of the ANS bits is required.

The trigger selected according to the ADTSR settings only enables activation of A/D conversion by group 0 or group 1.

19.4.4 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 19.6). 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 19.9): 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 19.7 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 19.7 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 19.6 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	—

A/D Converter Module	Analog Input Channels	Group
A/D converter module 2	AN8	—
	AN9	—
	AN10	—
	AN11	—
	AN12	—
	AN13	—
	AN14	—
	AN15	—

Table 19.7 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_{CNOV}	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)

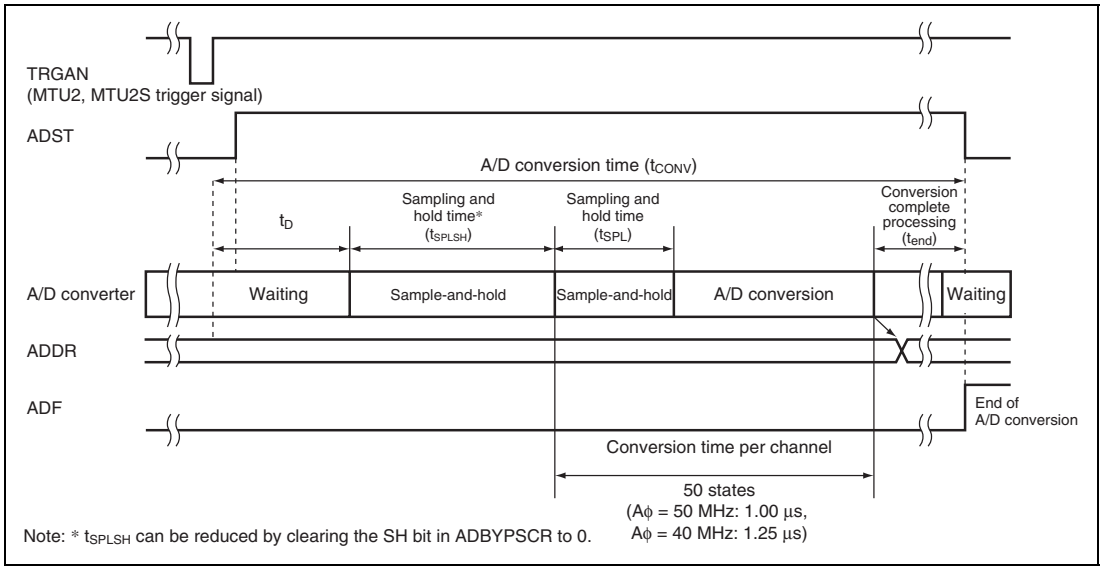


Figure 19.9 A/D Conversion Timing

19.4.5 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 in single-cycle scan mode or continuous scan mode, set the TRGE bit in ADCR to 1, clear the 2CHSE bit in ADTSR to 0, 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. To enable this function in 2-channel scan mode, set the TRGE bit in ADCR to 1 and set the 2CHSE bit in ADTSR to 1. After this setting is made, if an A/D conversion start trigger from the MTU2 or MTU2S specified by the TRG1S bit in ADTSR is generated, the ADST bit is set to 1.

19.4.6 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 clear the 2CHSE bit to 0 and set the EXTRG bit to 1 in ADTSR in single-cycle or continuous scan mode or set the 2CHSE bit to 1 and TRG0S or TRG1S bits to B'0000 in ADTSR in 2-channel scan mode. In this state, a trigger is input 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 19.10 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

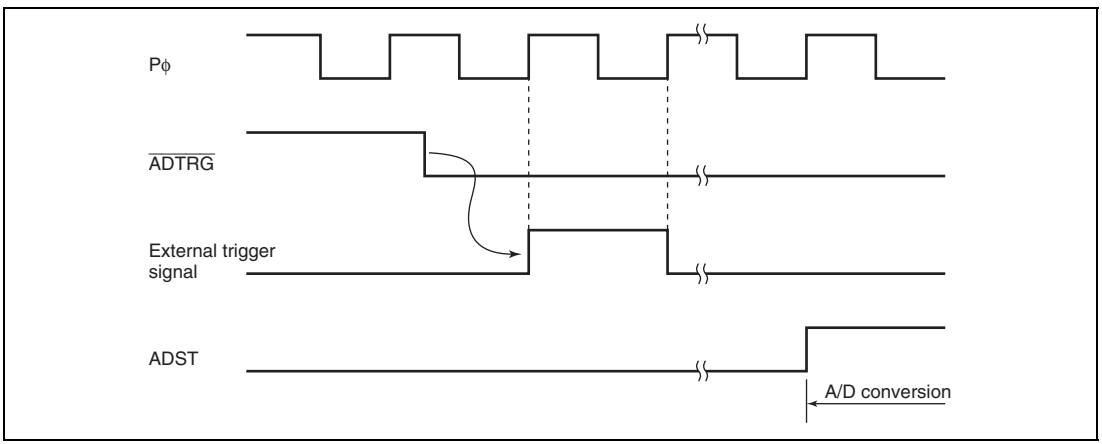


Figure 19.10 External Trigger Input Timing

19.4.7 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 19.11 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.

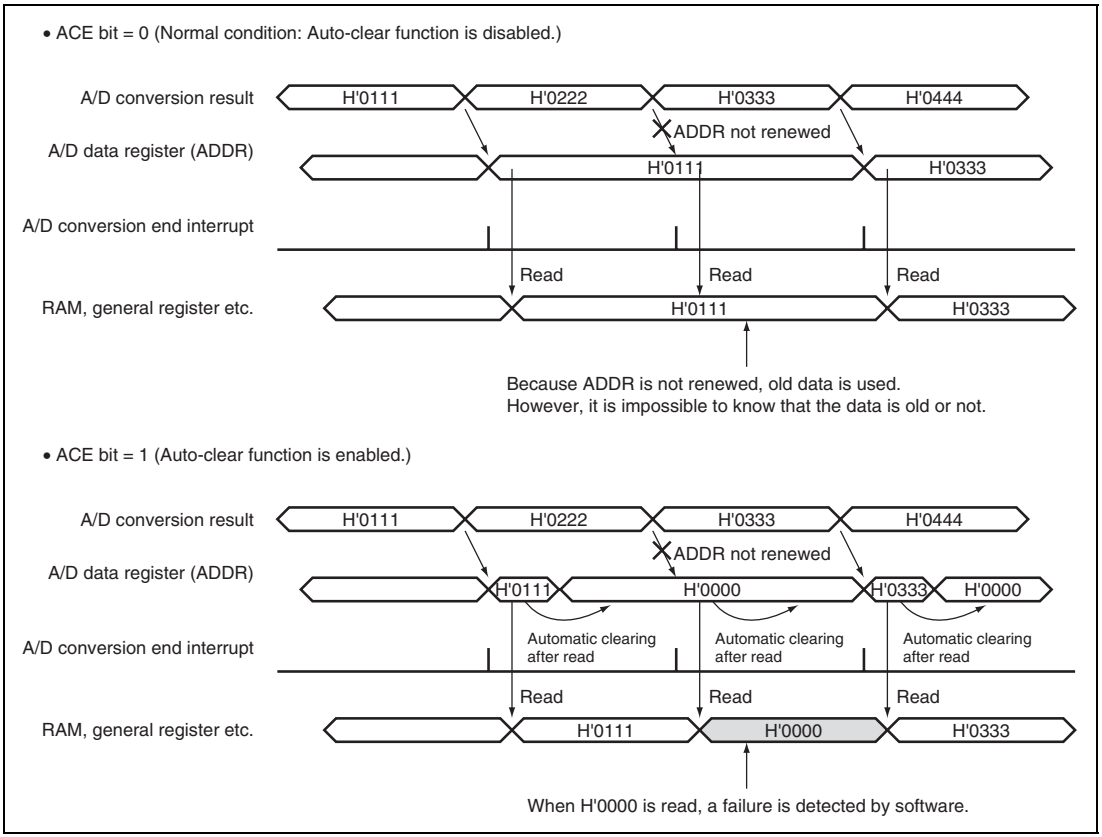


Figure 19.11 Example of When ADDR Auto-clear Function is Disabled (Normal Condition)/Enabled

19.4.8 A/D Conversion Synchronization Function

The A/D conversion synchronization function allows A/D_2 to be started synchronously with A/D_1. Since A/D_1 and A/D_2 share the analog power supply (AVCC), analog ground (AVSS), analog reference power supply (AVREF), and analog reference ground (AVREFVSS), deterioration of the conversion precision can be eliminated because using this function eliminates the noise caused by the difference of the conversion start timing between A/D_1 and A/D_2.

(1) Operation Example when A/D_1 and A/D_2 are in Single-Cycle Scan Mode

The following describes the operation example in which analog inputs 4 to 6 (AN4 to AN6) and analog inputs 8 to 10 (AN8 to AN10) are selected.

1. Clear the TRGE bit to 0 in A/D control register 2 (ADCR_2).
2. Set the ANS0 to ANS2 bits to 1 in A/D analog input channel select register 1 (ADANSR_1).
3. Set the ANS0 to ANS2 bits to 1 in A/D analog input channel select register 2 (ADANSR_2).
4. Set the ADSST bit to 1 in A/D bypass control register 2 (ADBYPSCR_2).
5. Set the ADST bit to 1 in A/D control register 1 (ADCR_1) to start A/D conversion by A/D_1.*
6. A/D conversion starts on channel 4. During the A/D conversion, the A/D synchronization start trigger signal is output. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR4.
7. The ADSST bit is automatically cleared to 0 by the A/D synchronization start trigger signal, thus setting the ADST bit in A/D control register 2 (ADCR_2). Sequentially, A/D conversion starts on channels 5 and 8 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR5 and ADDR8, respectively.
8. A/D conversion starts on channels 6 and 9 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR6 and ADDR9, respectively.
9. A/D_1 sets the ADF bit to 1 in A/D status register 1 (ADASR_1), automatically clearing the ADST bit to 0 in A/D control register 1 (ADCR_1), and stops conversion. Here, if the ADIE bit in ADCR_1 is 1, an ADI interrupt is generated on A/D conversion completion.
10. A/D conversion starts on channels 10. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR10.
11. A/D_2 sets the ADF bit to 1 in A/D status register 2 (ADASR_2), automatically clearing the ADST bit to 0 in A/D control register 2 (ADCR_2), and stops conversion. Here, if the ADIE bit in ADCR_2 is 1, an ADI interrupt is generated on A/D conversion completion.

Note: * If A/D_1 conversion is started during A/D_2 conversion, full synchronization will not be available; start A/D_1 while A/D_2 is in the idle state (ADST bit is 0 in A/D control register 2 (ADCR_2).

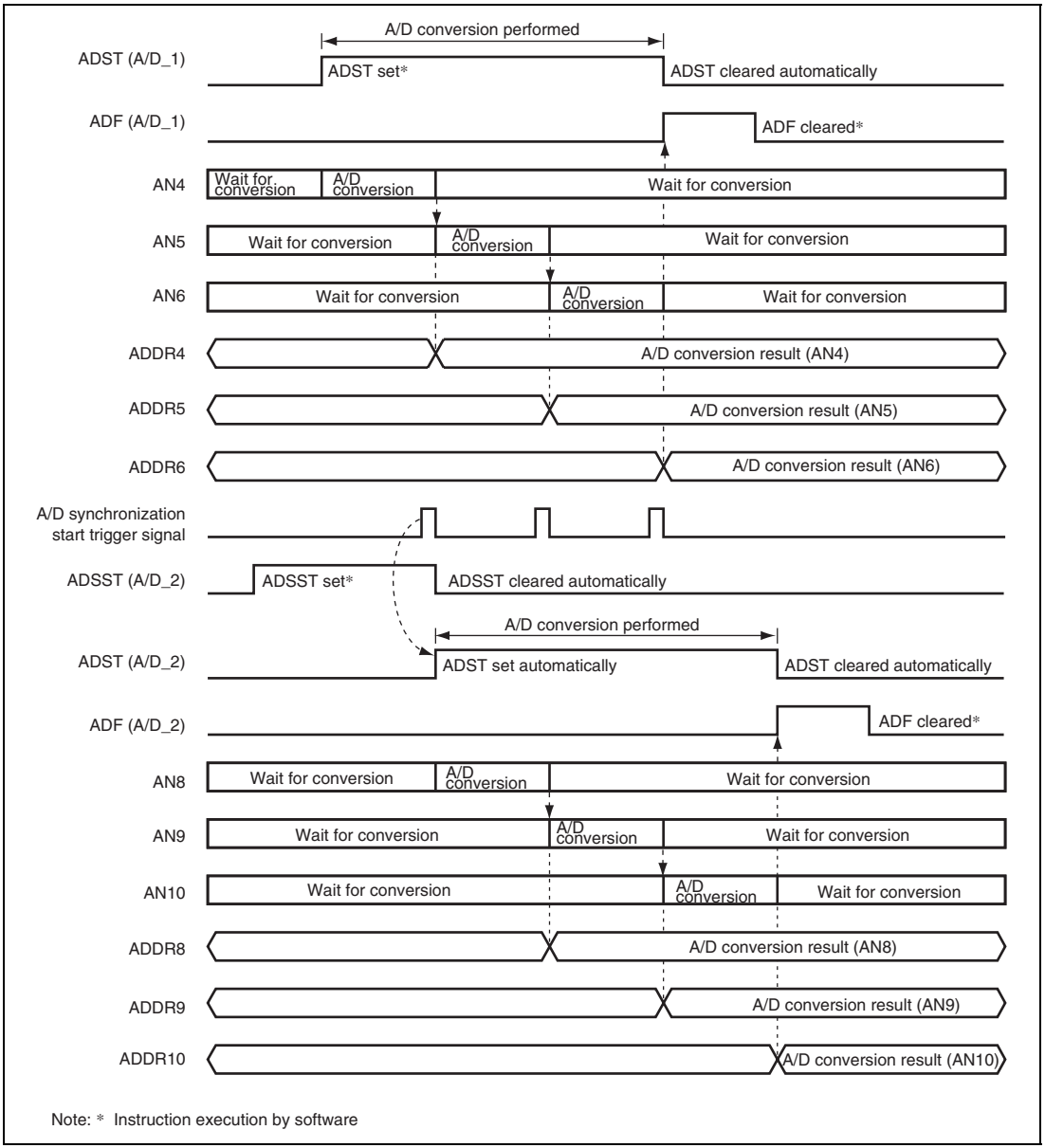


Figure 19.12 Operation Example when A/D_1 and A/D_2 are in Single-Cycle Scan Mode

(2) Operation Example when A/D_1 and A/D_2 are in Continuous Scan Mode

The following describes the operation example in which analog inputs 4 and 5 (AN4 and AN5) and analog inputs 9 and 10 (AN9 and AN10) are selected.

1. Clear the TRGE bit to 0 in A/D control register 2 (ADCR_2).
2. Set the ADCS bits to 1 in A/D control registers 1 and 2 (ADCR_1 and ADCR_2).
3. Set the ANS0 and ANS1 bits to 1 in A/D analog input channel select register 1 (ADANSR_1). *[1]
4. Set the ANS0 and ANS1 bits to 1 in A/D analog input channel select register 2 (ADANSR_2). *[1]
5. Set the ADSST bit to 1 in A/D bypass control register 2 (ADBYPSCR_2).
6. Set the ADST bit to 1 in A/D control register 1 (ADCR_1) to start A/D conversion by A/D_1.*[2]
7. A/D conversion starts on channel 4. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR4. Sequentially, A/D conversion starts on channel 5. During the A/D conversion, the A/D synchronization start trigger signal is output. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR5.
8. When the conversion on the specified channels (AN4 and AN5) is completed, A/D_1 sets the ADF bit to 1 in A/D status register 1 (ADASR_1). Here, if the ADIE bit in A/D control register 1 (ADCR_1) is 1, an ADI interrupt is generated on A/D conversion completion.
9. Steps 1 to 8 are repeated while the ADST bit in A/D control register 1 (ADCR_1) is 1.
10. The ADSST bit is automatically cleared to 0 by the A/D synchronization start trigger signal, thus setting the ADST bit in A/D control register 2 (ADCR_2).
11. A/D conversion starts on channels 4 and 8 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR4 and ADDR8, respectively. Sequentially, A/D conversion starts on channels 5 and 9 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR5 and ADDR9, respectively.
12. When the conversion on the specified channels (AN8 and AN9) is completed, A/D_2 sets the ADF bit to 1 in A/D status register 2 (ADASR_2). Here, if the ADIE bit in A/D control register 2 (ADCR_2) is 1, an ADI interrupt is generated on A/D conversion completion
13. Steps 11 and 12 are repeated while the ADST bit in A/D control register 2 (ADCR_2) is 1. Clearing the ADST bit to 0 stops A/D conversion.*[3]

Notes: 1. Select the same number of channels for A/D_1 and A/D_2. Otherwise, full synchronization will not be available.

2. If A/D_1 conversion is started during A/D_2 conversion, full synchronization will not be available; start A/D_1 while A/D_2 is in the idle state (ADST bit is 0 in A/D control register 2 (ADCR_2).
3. A/D_2 continues conversion as long as the ADST bit in A/D control register 2 (ADCR_2) is 1, irrespective of whether A/D_1 is operating or not.

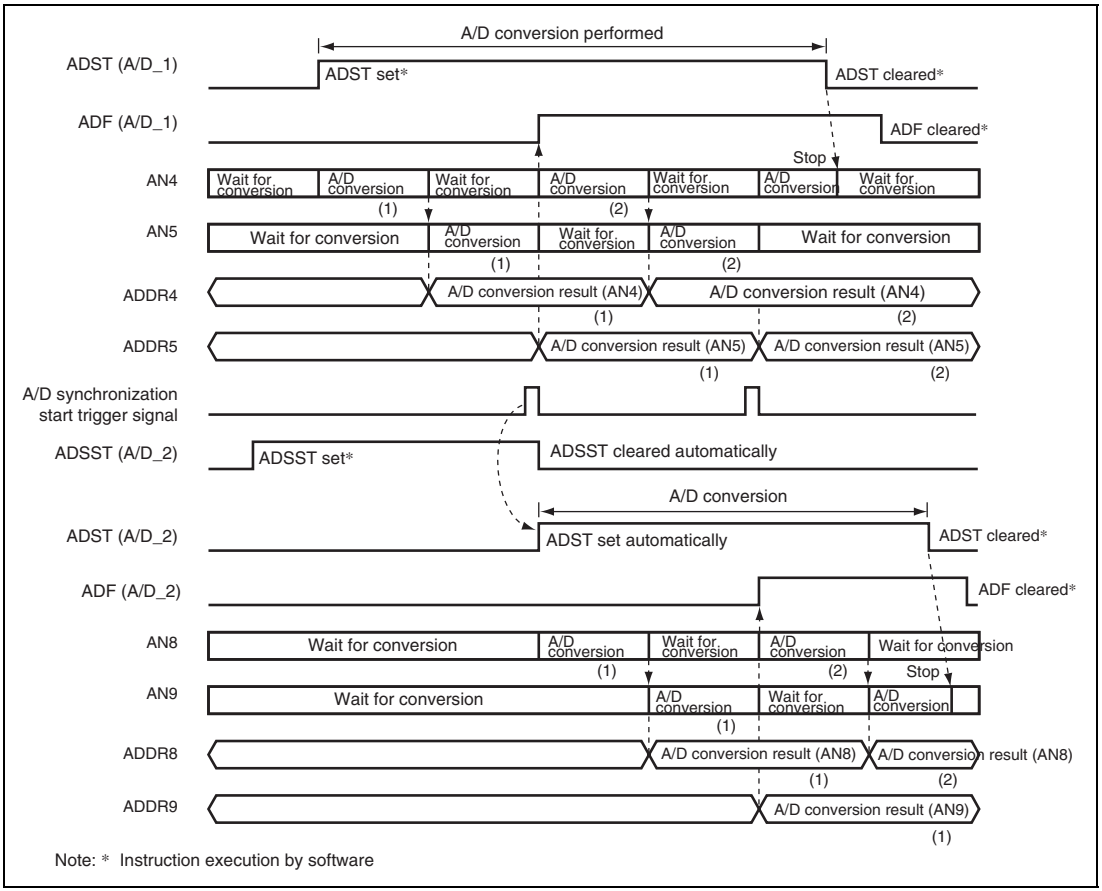


Figure 19.13 Operation Example when A/D_1 and A/D_2 are in Continuous Scan Mode

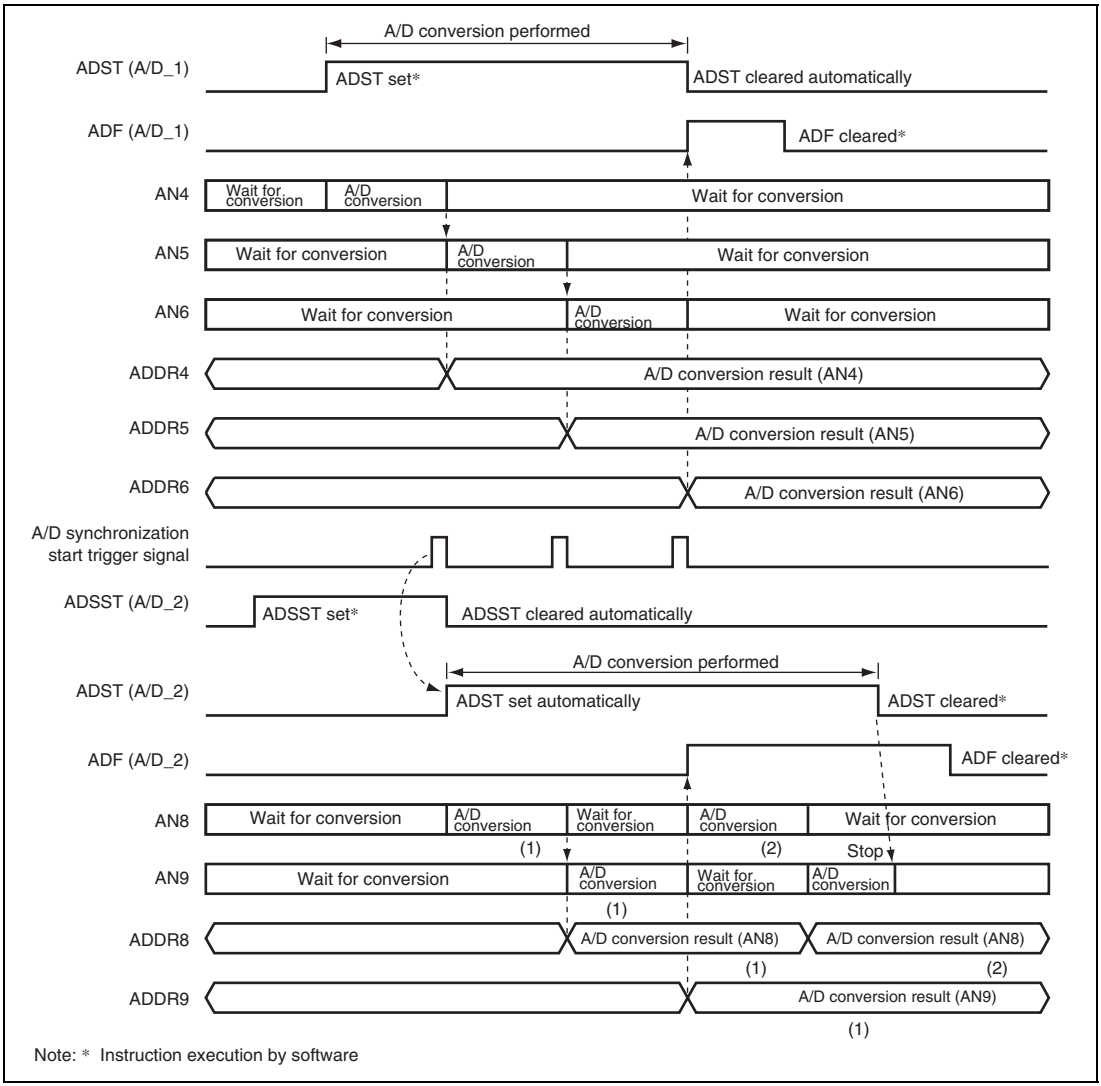
(3) Operation Example when A/D_1 is in Single-Cycle Scan Mode and A/D_2 is in Continuous Scan Mode

The following describes the operation example in which analog inputs 4 to 6 (AN4 and AN6) and analog inputs 8 and 9 (AN8 and AN9) are selected.

1. Clear the TRGE bit to 0 in A/D control register 2 (ADCR_2).
2. Set the ADCS bit to 1 in A/D control register 1 (ADCR_1).
3. Set the ANS0 to ANS2 bits to 1 in A/D analog input channel select register 1 (ADANSR_1).*[1]
4. Set the ANS0 to ANS1 bits to 1 in A/D analog input channel select register 2 (ADANSR_2).*[1]
5. Set the ADSST bit to 1 in A/D bypass control register 2 (ADBYPSCR_2).
6. Set the ADST bit to 1 in A/D control register 1 (ADCR_1) to start A/D conversion by A/D_1.*[2]
7. A/D conversion starts on channel 4. During the A/D conversion, the A/D synchronization start trigger signal is output. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR4.
8. The ADSST bit is automatically cleared to 0 by the A/D synchronization start trigger signal, thus setting the ADST bit in A/D control register 2 (ADCR_2).
9. A/D conversion starts on channels 5 and 8 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR5 and ADDR8, respectively.
10. A/D conversion starts on channels 6 and 9 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR6 and ADDR9, respectively.
11. A/D_1 sets the ADF bit to 1 in A/D status register 1 (ADASR_1), automatically clearing the ADST bit to 0 in A/D control register 1 (ADCR_1), and stops conversion. Here, if the ADIE bit in ADCR_2 is 1, an ADI interrupt is generated on A/D conversion completion.
12. When the conversion on the specified channels (AN8 and AN9) is completed, A/D_2 sets the ADF bit to 1 in A/D status register 2 (ADASR_2). Here, if the ADIE bit in A/D control register 2 (ADCR_2) is 1, an ADI interrupt is generated on A/D conversion completion
13. Steps 9, 10 and 12 are repeated while the ADST bit in A/D control register 2 (ADCR_2) is 1. Clearing the ADST bit to 0 stops A/D conversion.*[3]

Notes: 1. Select the smaller number of channels for A/D_2 than for A/D_1. Otherwise, full synchronization will not be available.
2. If A/D_1 conversion is started during A/D_2 conversion, full synchronization will not be available; start A/D_1 while A/D_2 is in the idle state (ADST bit is 0 in A/D control register 2 (ADCR_2).

3. A/D_2 continues conversion as long as the ADST bit in A/D control register 2 (ADCR_2) is 1, irrespective of whether A/D_1 is operating or not.



Note: * Instruction execution by software

Figure 19.14 Operation Example when A/D_1 is in Single-Cycle Scan Mode and A/D_2 is in Continuous Scan Mode

(4) Operation Example when A/D_1 is in Continuous Scan Mode and A/D_2 is in Single-Cycle Scan Mode

The following describes the operation example in which analog inputs 4 and 5 (AN4 and AN5) and analog inputs 8 and 9 (AN8 and AN9) are selected.

1. Clear the TRGE bit to 0 in A/D control register 2 (ADCR_2).
2. Set the ADCS bit to 1 in A/D control register 1 (ADCR_1).
3. Set the ANS0 and ANS1 bits to 1 in A/D analog input channel select register 1 (ADANSR_1).*[1]
4. Set the ANS0 to ANS1 bits to 1 in A/D analog input channel select register 2 (ADANSR_2).*[1]
5. Set the ADSST bit to 1 in A/D bypass control register 2 (ADBYPSCR_2).
6. Set the ADST bit to 1 in A/D control register 1 (ADCR_1) to start A/D conversion by A/D_1.*[2]
7. A/D conversion starts on channel 4. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR4. During the A/D conversion, the A/D synchronization start trigger signal is output. When the A/D conversion is completed, the result of the A/D conversion is transferred to ADDR5.
8. When the conversion on the specified channels (AN4 and AN5) is completed, A/D_1 sets the ADF bit to 1 in A/D status register 1 (ADASR_1). Here, if the ADIE bit in A/D control register 1 (ADCR_1) is 1, an ADI interrupt is generated on A/D conversion completion.
9. Steps 7 and 8 are repeated while the ADST bit in A/D control register 1 (ADCR_1) is 1.
10. The ADSST bit is automatically cleared to 0 by the A/D synchronization start trigger signal, thus setting the ADST bit in A/D control register 2 (ADCR_2).
11. A/D conversion starts on channels 4 and 8 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR4 and ADDR8, respectively. Sequentially, A/D conversion starts on channels 5 and 9 synchronously. When the A/D conversion is completed, the results of the A/D conversion are transferred to ADDR5 and ADDR9, respectively.
12. A/D_2 sets the ADF bit to 1 in A/D status register 2 (ADASR_2), automatically clearing the ADST bit to 0 in A/D control register 2 (ADCR_2), and stops conversion. Here, if the ADIE bit in ADCR_2 is 1, an ADI interrupt is generated on A/D conversion completion.

Notes: 1. Select the smaller number of channels for A/D_2 than for A/D_1. Otherwise, full synchronization will not be available.
2. If A/D_1 conversion is started during A/D_2 conversion, full synchronization will not be available; start A/D_1 while A/D_2 is in the idle state (ADST bit is 0 in A/D control register 2 (ADCR_2).

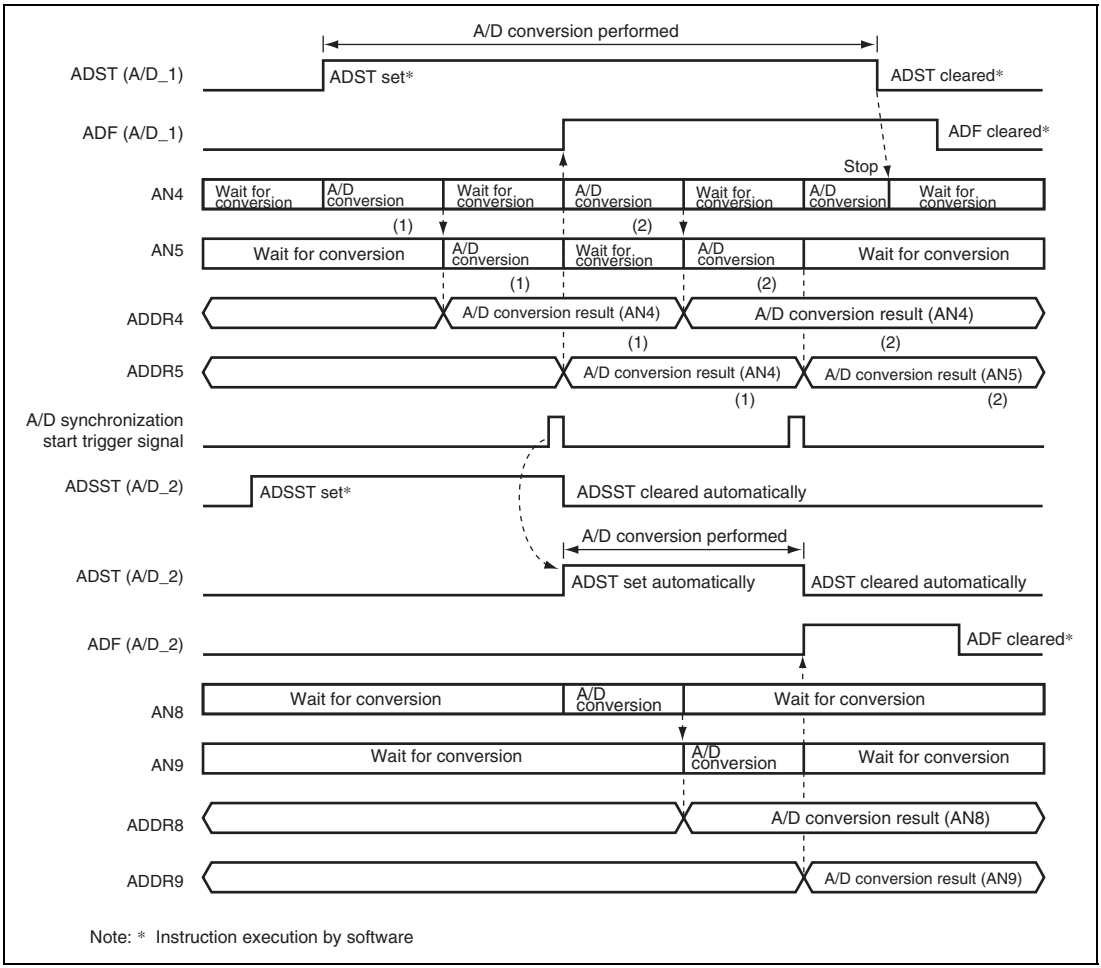


Figure 19.15 Operation Example when A/D_1 is in Continuous Scan Mode and A/D_2 is in Single-Cycle Scan Mode

(5) Operation Example when A/D_1 is Forcibly Terminated

If the ADST bit in A/D control register 1 (ADCR_1) is cleared to 0 before A/D_1 outputs the A/D synchronization start trigger signal to terminate A/D conversion, A/D_2 does not start conversion. If A/D_1 starts conversion again and outputs the A/D synchronization start trigger signal, A/D_2 starts conversion.

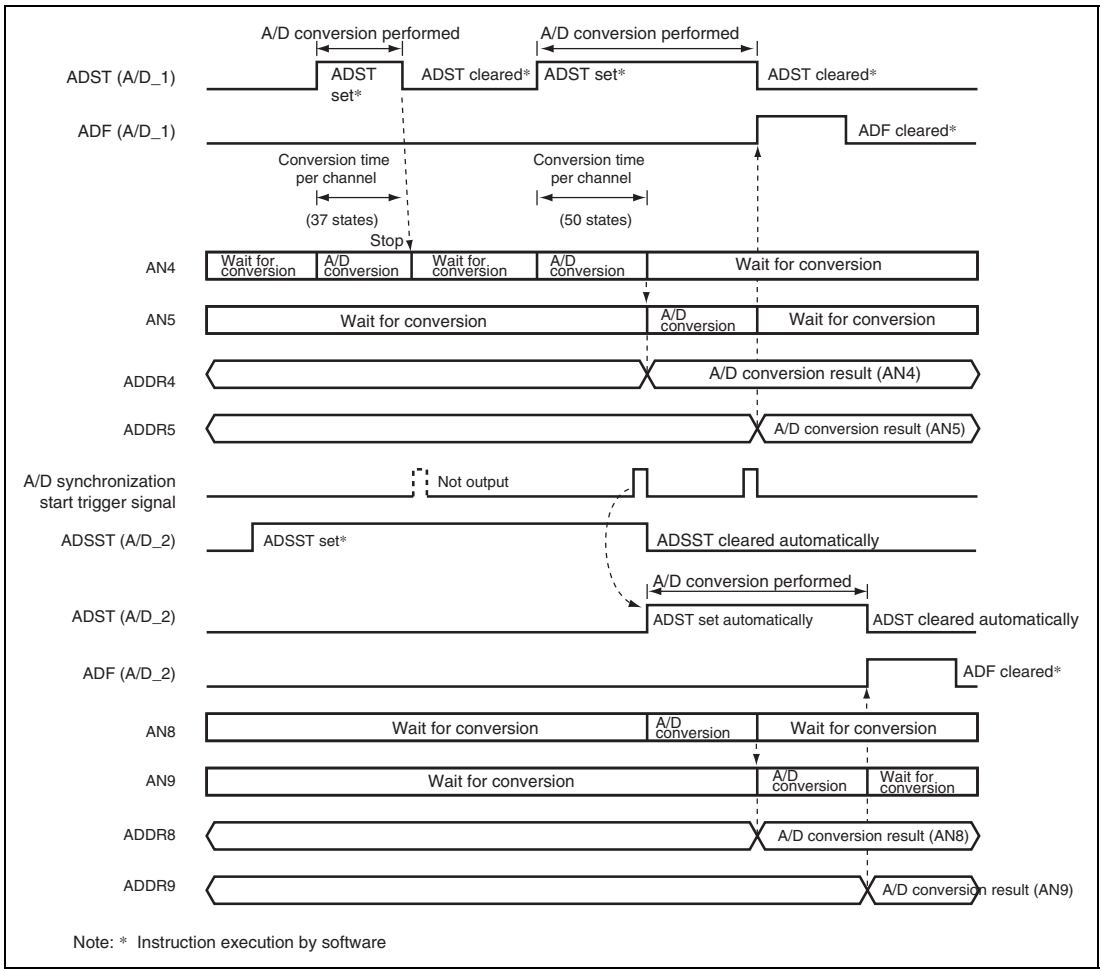
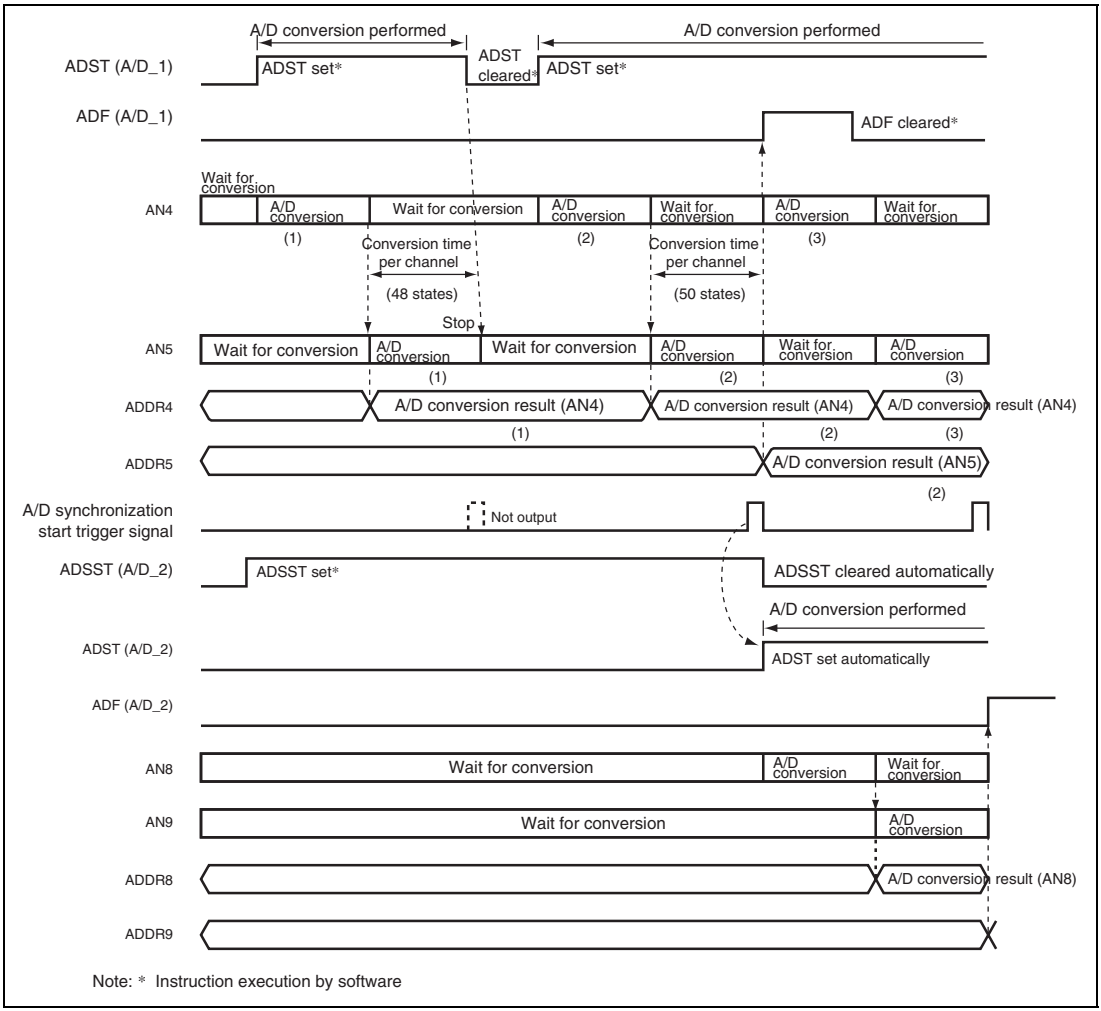


Figure 19.16 Operation Example when A/D_1 is in Single-Cycle Scan Mode



Note: * Instruction execution by software

Figure 19.17 Operation Example when A/D_1 is in Continuous Scan Mode

19.5 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 19.8 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
A/D converter module 2	ADI2	Not available	Available

19.6 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 19.18).
- 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 19.18).
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 19.18).
- 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 19.18).
- Absolute accuracy

  The deviation between the digital value and the analog input value. Includes offset error, full-scale error, quantization error, and nonlinearity error.

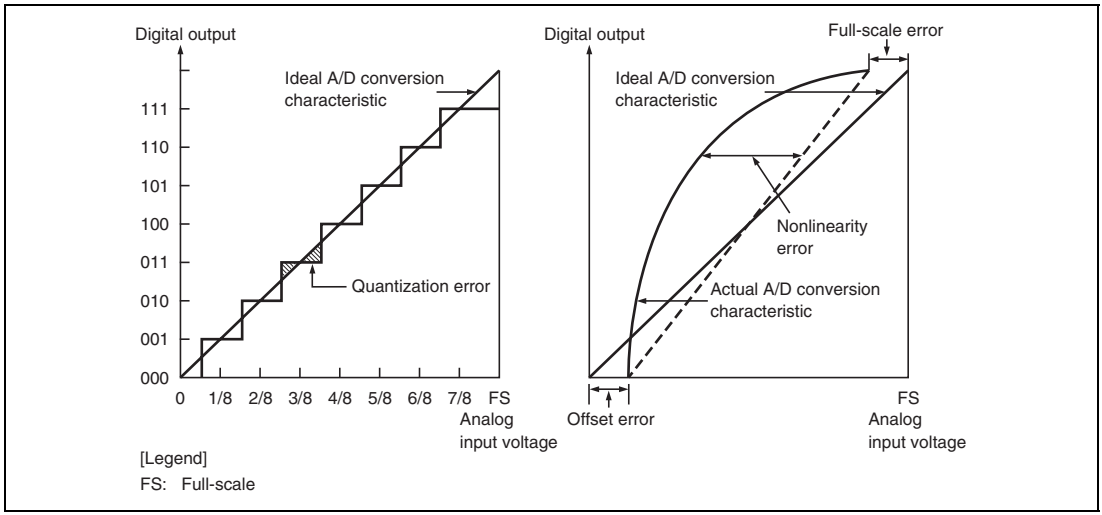


Figure 19.18 Definitions of A/D Conversion Accuracy

19.7 Usage Notes

19.7.1 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 15) ≤ AVREF.

19.7.2 Relationship between AVCC, AVSS and VCC, VSS

When using the A/D converter, set VCC ≤ AVCC = 5.0 V ± 0.5 V and AVSS = VSS. When the A/D converter is not used, set VCC ≤ AVCC ≤ 5.0V ± 0.5 V, AVSS = VSS, and do not leave the AVCC pin open.

19.7.3 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.

19.7.4 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 AN15), 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.

19.7.5 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN15) and analog reference power supply (AVREF), a protection circuit should be connected between the AVCC and AVSS, as shown in figure 19.19. 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 19.19 should be placed close to the pin.

If a filter capacitor is connected as shown in figure 19.19, 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.

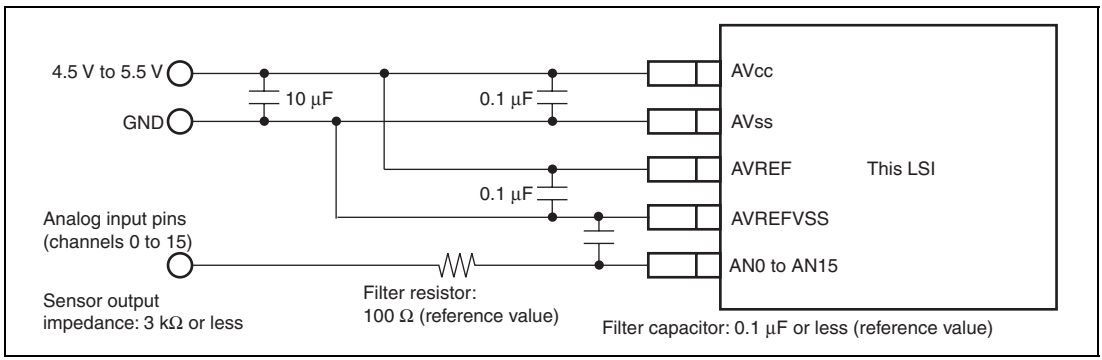


Figure 19.19 Example of Analog Input Pin Protection Circuit

19.7.6 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 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.

19.7.7 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).

19.7.8 Notes when Two or More A/D Modules Run Simultaneously

This LSI has three A/D modules. When two or more 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 19.20 to 19.22, the guaranteed absolute precision of the A/D conversion module which has been activated first will be the values as listed in tables 19.9 to 19.11. The absolute precision depends on the cycle difference ($T_{AD0\text{-}AD1}$ in figures 19.20 and 19.21 and $T_{AD2\text{-}AD1}$ in figure 19.22) 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.

Two power supplies are provided for the A/D modules. One is for A/D_0 only and the other is for A/D_1 and A/D_2. If A/D_1 and A/D_2 run simultaneously, the interference is increased because the same power supply is used.

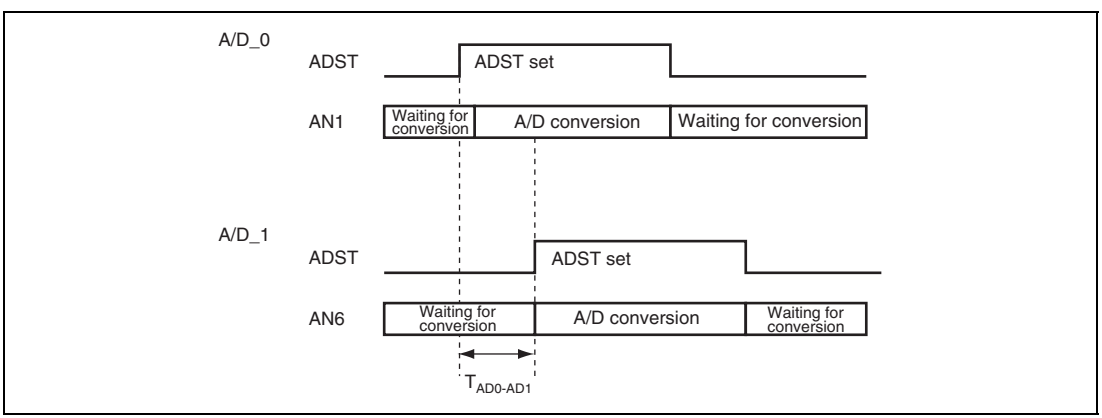


Figure 19.20 A/D Conversion Start Timing between A/D_0 Converter and A/D_1 Converter (Sample-and-Hold Circuits Disabled in A/D_0 and A/D_1) (Interference between Units Which Use Different Power Supplies)

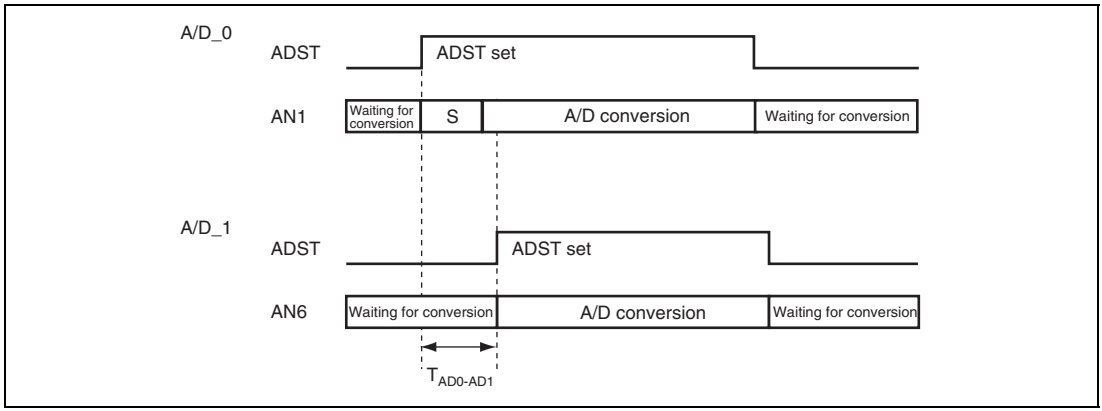


Figure 19.21 A/D Conversion Start Timing between A/D_0 Converter and A/D_1 Converter (Sample-and-Hold Circuit Enabled in A/D_0)

Table 19.9 Absolute Precision and A/D Conversion Start Cycle Difference, $T_{AD0\text{-}AD1}$ (Aϕ) between A/D_0 and A/D_1 in Figure 19.20 (Interference between Units Which Use Different Power Supplies)

	$T_{AD0\text{-}AD1}$	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. 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 $T_{AD0\text{-}AD1} = 0$.
3. When two A/D modules run simultaneously, the absolute precision of the first activated A/D is not guaranteed except for $T_{AD0\text{-}AD1}$.
4. 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 19.10 Absolute Precision and A/D Conversion Start Cycle Difference, $T_{AD0\text{-}AD1}$ (Aϕ) between A/D_0 and A/D_1 in Figure 19.21

	$T_{AD0\text{-}AD1}$	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. 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 $T_{AD0\text{-}AD1} = 0$.
3. When two A/D modules run simultaneously, the absolute precision of the first activated A/D is not guaranteed except for $T_{AD0\text{-}AD1}$.
4. 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.

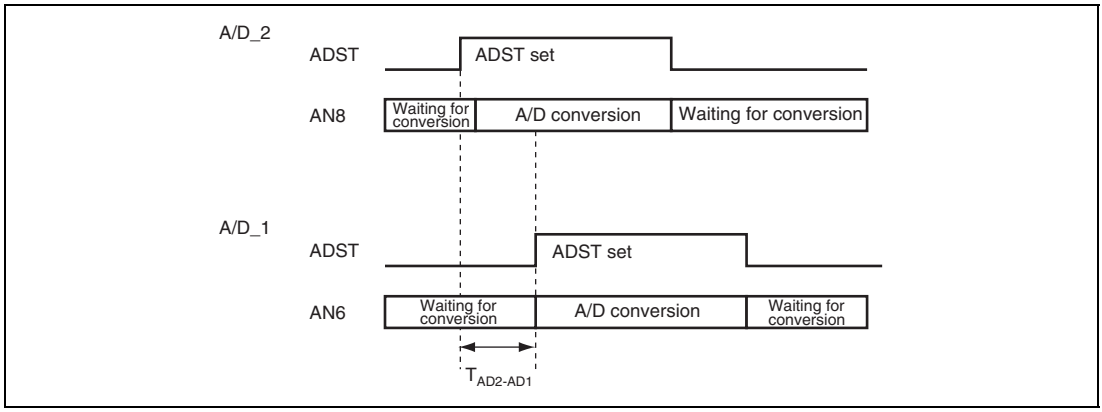


Figure 19.22 A/D Conversion Start Timing between A/D_2 Converter and A/D_1 Converter (Interference between Units Which Use Same Power Supply)

Table 19.11 Absolute Precision and A/D Conversion Start Cycle Difference, $T_{AD2-AD1}$ (Aϕ) between A/D_2 and A/D_1 in Figure 19.22 (Interference between Units Which Use Same Power Supply)

	$T_{AD0-AD1}$	Unit
	0 to 14, 27 to 30, 48 or more	**Aϕ (clock)**
Absolute precision	±8	LSB

Notes: 1. This table lists the A/D_2 absolute precision when the converter of A/D_2 is started first.
2. When the conversion of A/D_2 and A/D_1 is started simultaneously, the absolute precision values of A/D_0 and A/D_1 are ±8LSB because $T_{AD2-AD1} = 0$.
3. When two A/D modules run simultaneously, the absolute precision of the first activated A/D is not guaranteed except for $T_{AD2-AD1}$.
4. When A/D_2 and A/D_1 are activated separately, each of $T_{AD2-AD1}$ values is 45 or more. Thus, the absolute precision values of A/D_2 and A/D_1 are ±8LSB.

Section 20 Controller Area Network (RCAN-ET)

20.1 Summary

20.1.1 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.

20.1.2 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.

20.1.3 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.

20.1.4 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)

20.1.5 Features

- supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 16 Mailbox version
- Clock 20 to 50 MHz or 20 to 40 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

20.2 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.

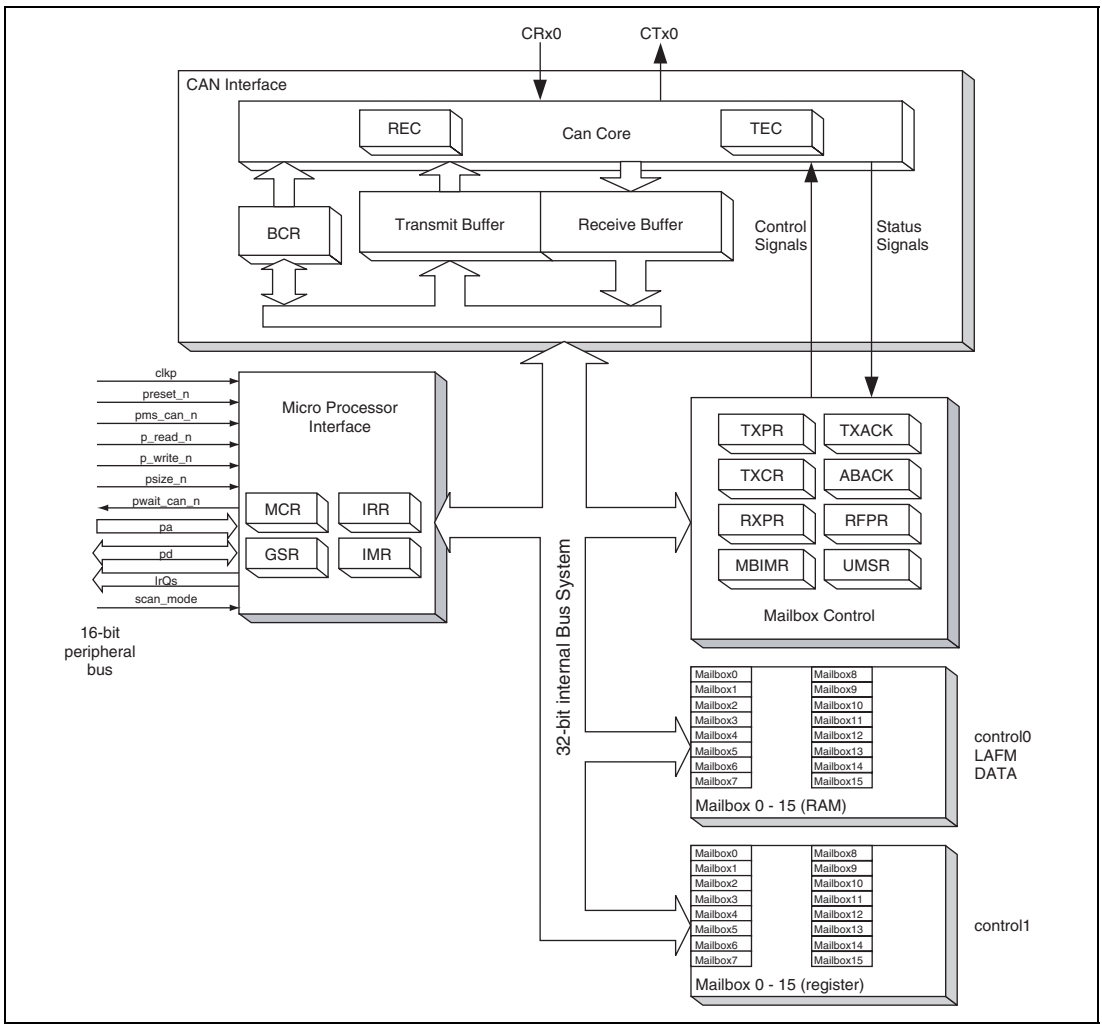


Figure 20.1 RCAN-ET Architecture

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.

20.3 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.

20.3.1 Memory Map

The diagram of the memory map is shown below. Register addresses in the figure below are offset addresses relative to H'FFFFD000.

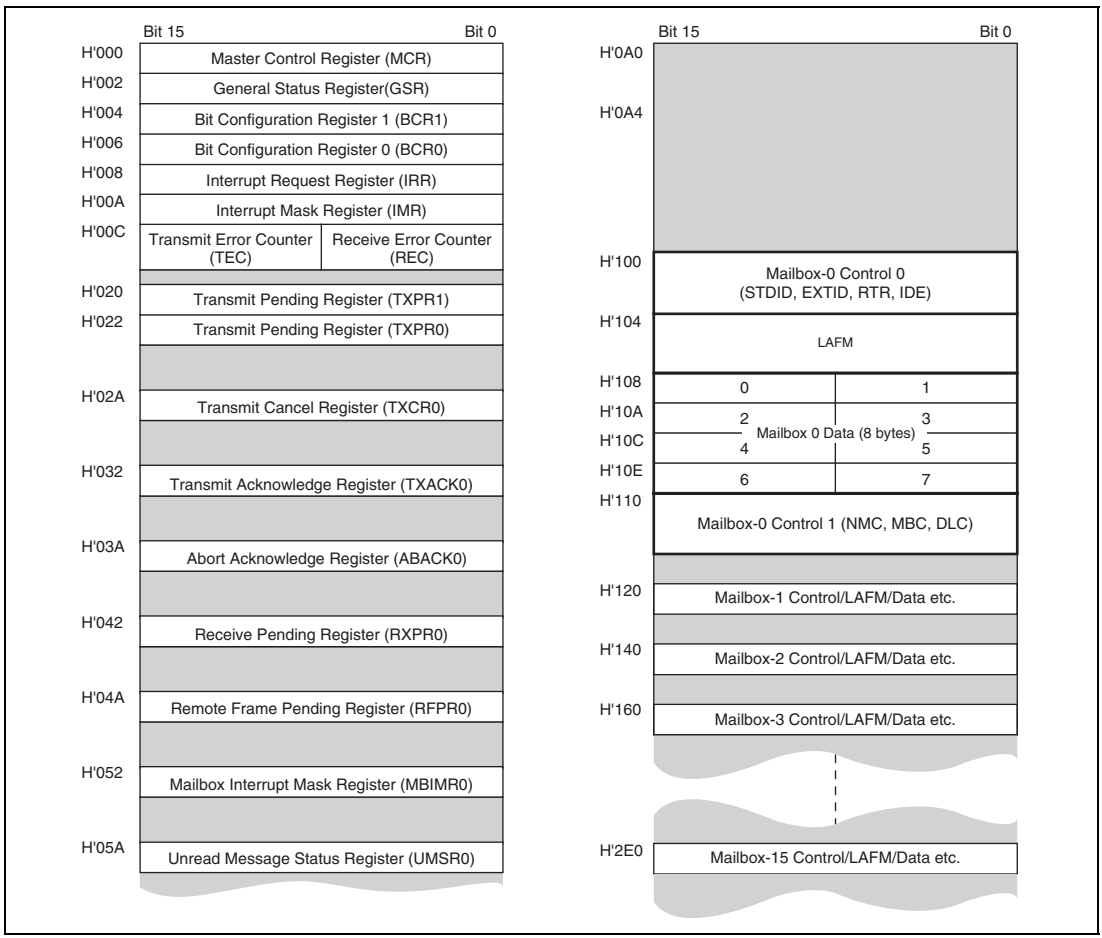


Figure 20.2 RCAN-ET Memory Map

The locations not used (between H'000 and H'2F2) are reserved and cannot be accessed.

20.3.2 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 20.1 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 20.3 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.

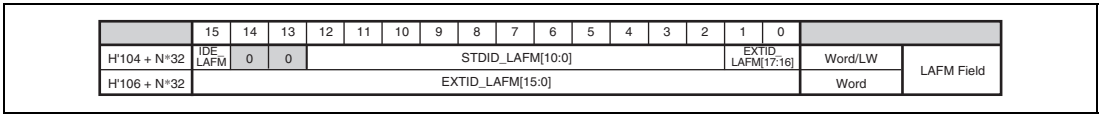


Figure 20.4 Acceptance Filter

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.

20.3.3 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	H'000	MCR	16
General Status Register	H'002	GSR	16
Bit Configuration Register 1	H'004	BCR1	16
Bit Configuration Register 0	H'006	BCR0	16
Interrupt Request Register	H'008	IRR	16
Interrupt Mask Register	H'00A	IMR	16
Error Counter Register	H'00C	TEC/REC	16

Figure 20.5 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)

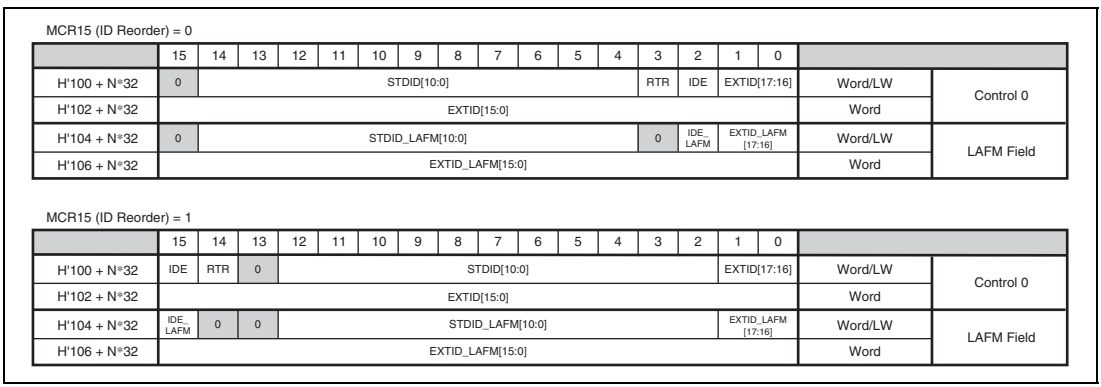


Figure 20.6 ID Reorder

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 × 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 20.4.1, 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)

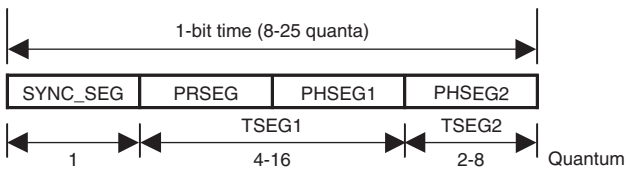
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 X peripheral bus clock (Initial value)
0	0	0	0	0	0	0	1	4 X peripheral bus clock
0	0	0	0	0	0	1	0	6 X peripheral bus clock
: :	: :	: :	: :	: :	: :	: :	: :	2*(register value+1) X peripheral bus clock
1	1	1	1	1	1	1	1	512 X 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 (resynchronisation) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronisation (resynchronisation) is established)

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

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 20.9.

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.

20.3.4 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.

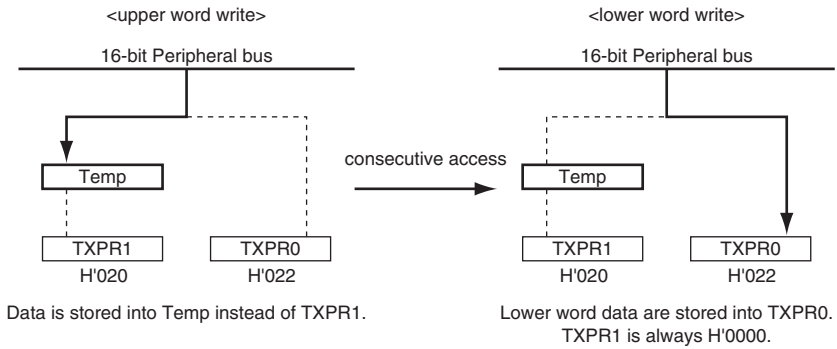
Description	Address	Name	Access Size (bits)
Transmit Pending 1	H'FFFFD020	TXPR1	16, 32
Transmit Pending 0	H'FFFFD022	TXPR0	—
	H'FFFFD024		
	H'FFFFD026		
	H'FFFFD028		
Transmit Cancel 0	H'FFFFD02A	TXCR0	
	H'FFFFD02C		
	H'FFFFD02E		
	H'FFFFD030		
Transmit Acknowledge 0	H'FFFFD032	TXACK0	16
	H'FFFFD034		
	H'FFFFD036		
	H'FFFFD038		
Abort Acknowledge 0	H'FFFFD03A	ABACK0	16
	H'FFFFD03C		
	H'FFFFD03E		
	H'FFFFD040		
Data Frame Receive Pending 0	H'FFFFD042	RXPR0	16
	H'FFFFD044		
	H'FFFFD046		
	H'FFFFD048		
Remote Frame Receive Pending 0	H'FFFFD04A	RFPR0	16
	H'FFFFD04C		
	H'FFFFD04E		
	H'FFFFD050		
Mailbox Interrupt Mask Register 0	H'FFFFD052	MBIMR0	16
	H'FFFFD054		
	H'FFFFD056		
	H'FFFFD058		
Unread message Status Register 0	H'FFFFD05A	UMSR0	16
	H'FFFFD05C		
	H'FFFFD05E		

Figure 20.7 RCAN-ET Mailbox Registers

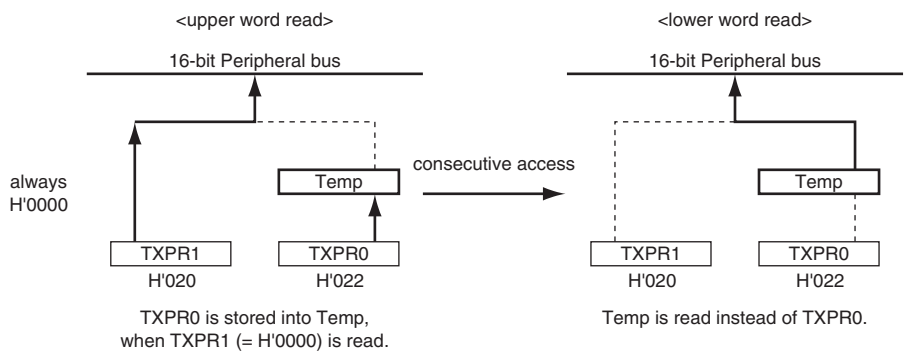
(1) Transmit Pending Register (TXPR1, TXPR0)

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 20.4, 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.

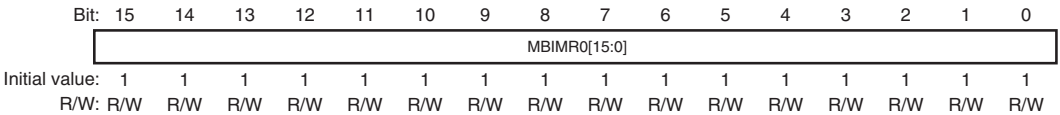
Bit[15:0]: RFPR0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received Remote Frame [Setting Condition] Completion of remote frame receive in corresponding mailbox

(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

20.4 Application Note

20.4.1 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.

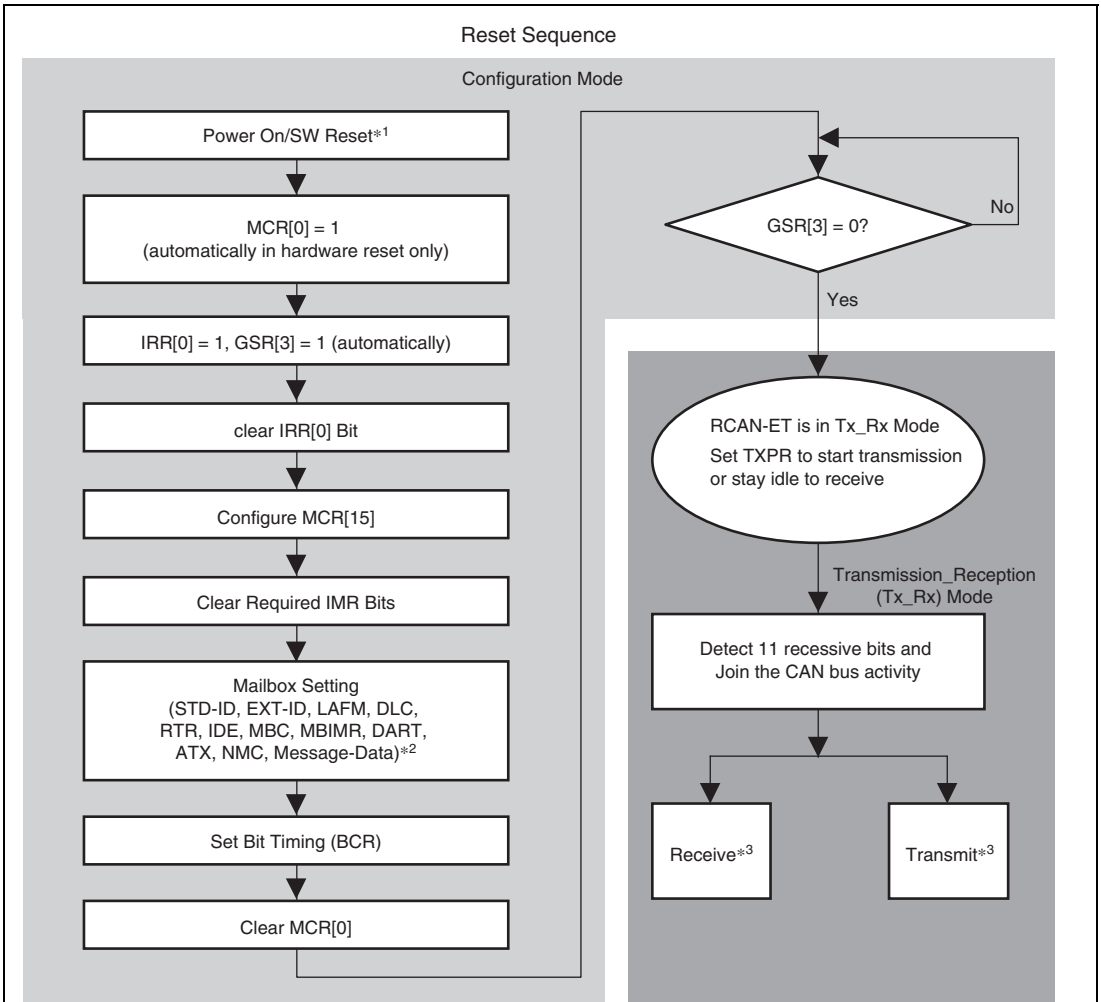
Note: the REC will not be modified by implementing this Mode. However, once running in Error Passive Mode, the REC will increase normally should errors be received. In this Mode, RCAN-ET will enter BusOff if TEC reaches 256 (Dec). However when this mode is used RCAN-ET will not be able to become Error 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.

20.4.2 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 20.8 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.

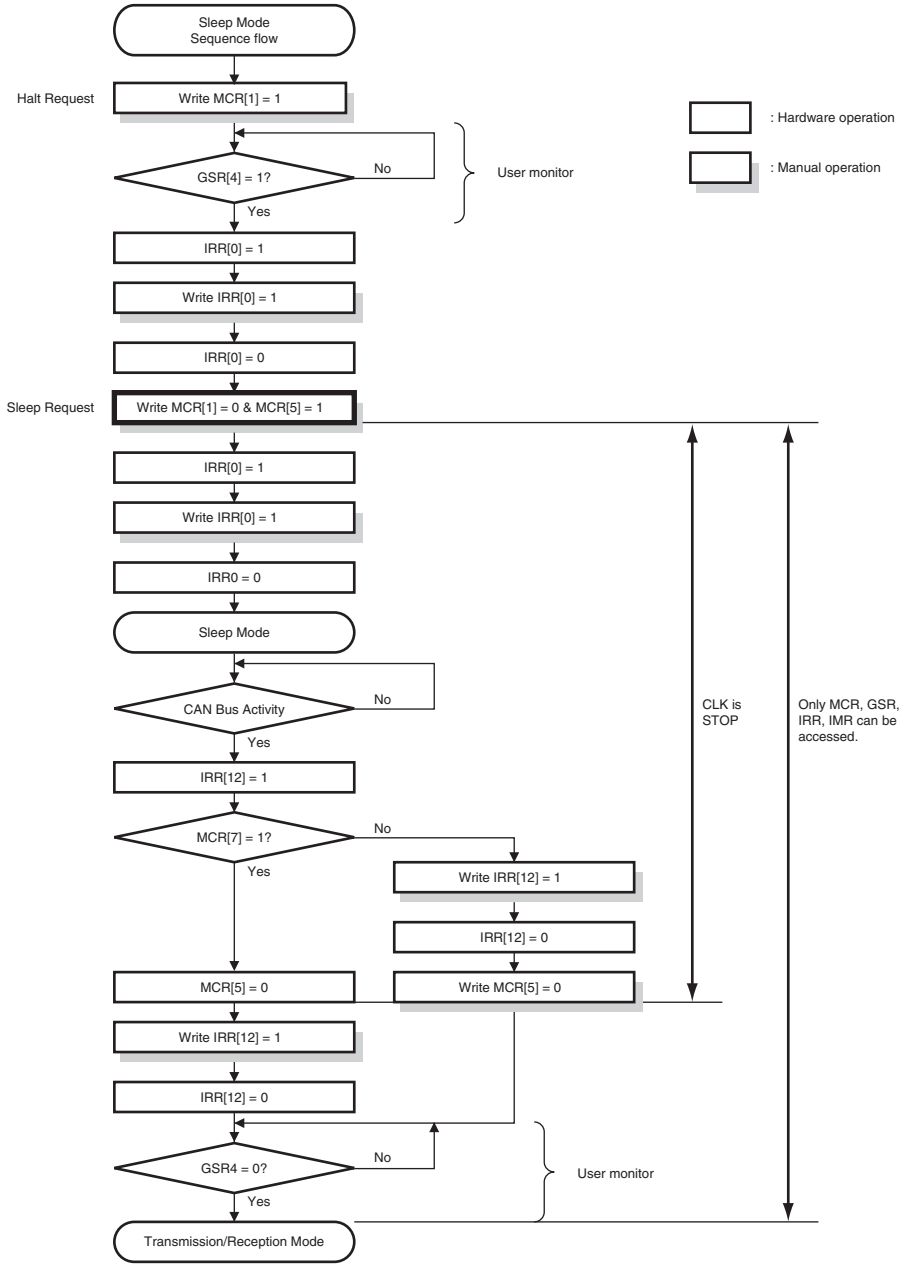

Sleep Mode Sequence flow
Halt Request
Write MCR[1] = 1
GSR[4] = 1?
No
Yes
User monitor
: Hardware operation
: Manual operation
IRR[0] = 1
Write IRR[0] = 1
IRR[0] = 0
Sleep Request
Write MCR[1] = 0 & MCR[5] = 1
IRR[0] = 1
Write IRR[0] = 1
IRR0 = 0
Sleep Mode
CAN Bus Activity
No
Yes
IRR[12] = 1
MCR[7] = 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
CLK is STOP
Only MCR, GSR, IRR, IMR can be accessed.

Figure 20.9 - 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.

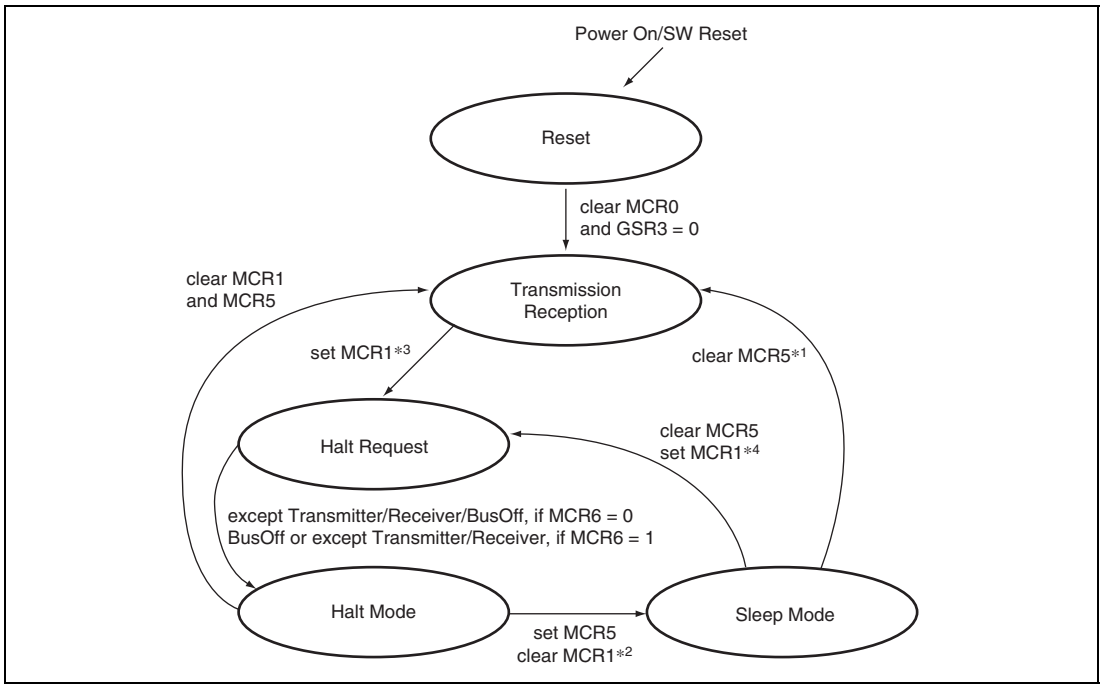


Figure 20.9 Halt Mode / Sleep Mode

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.

20.4.3 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).

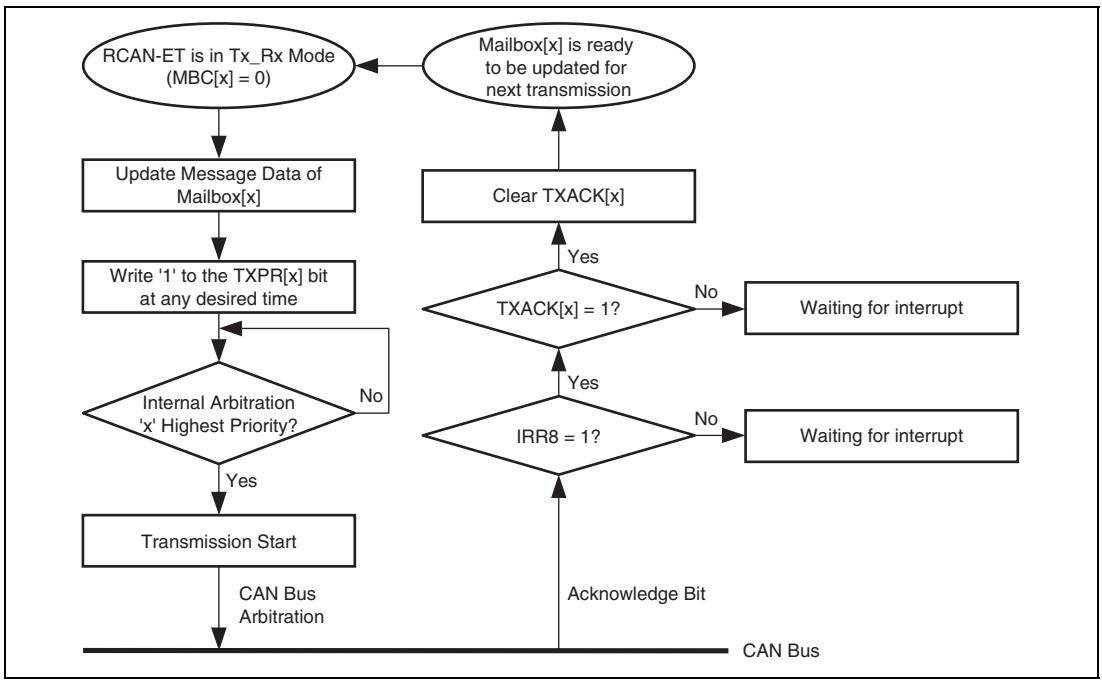


Figure 20.10 Transmission Request

- 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.

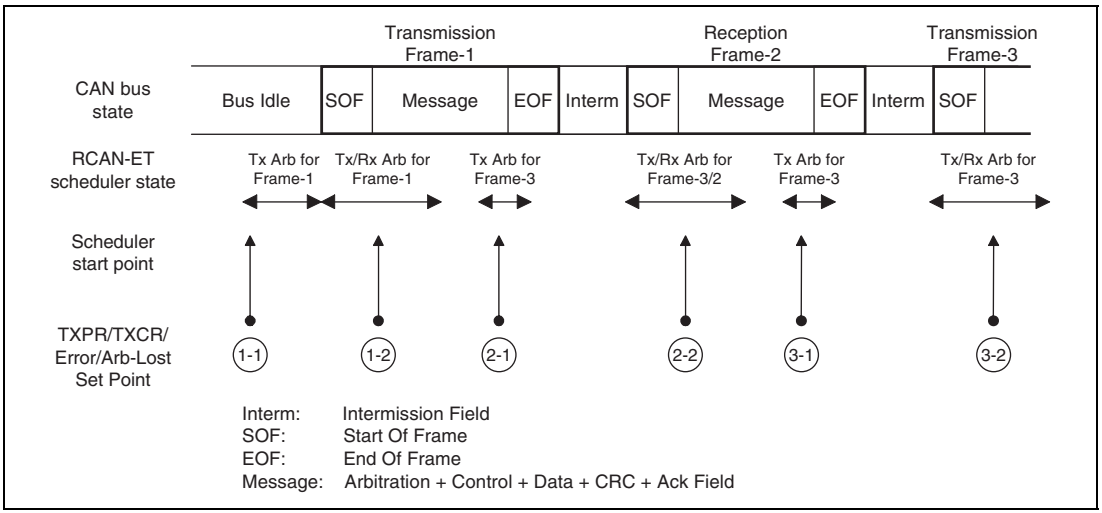


Figure 20.11 Internal Arbitration for Transmission

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.

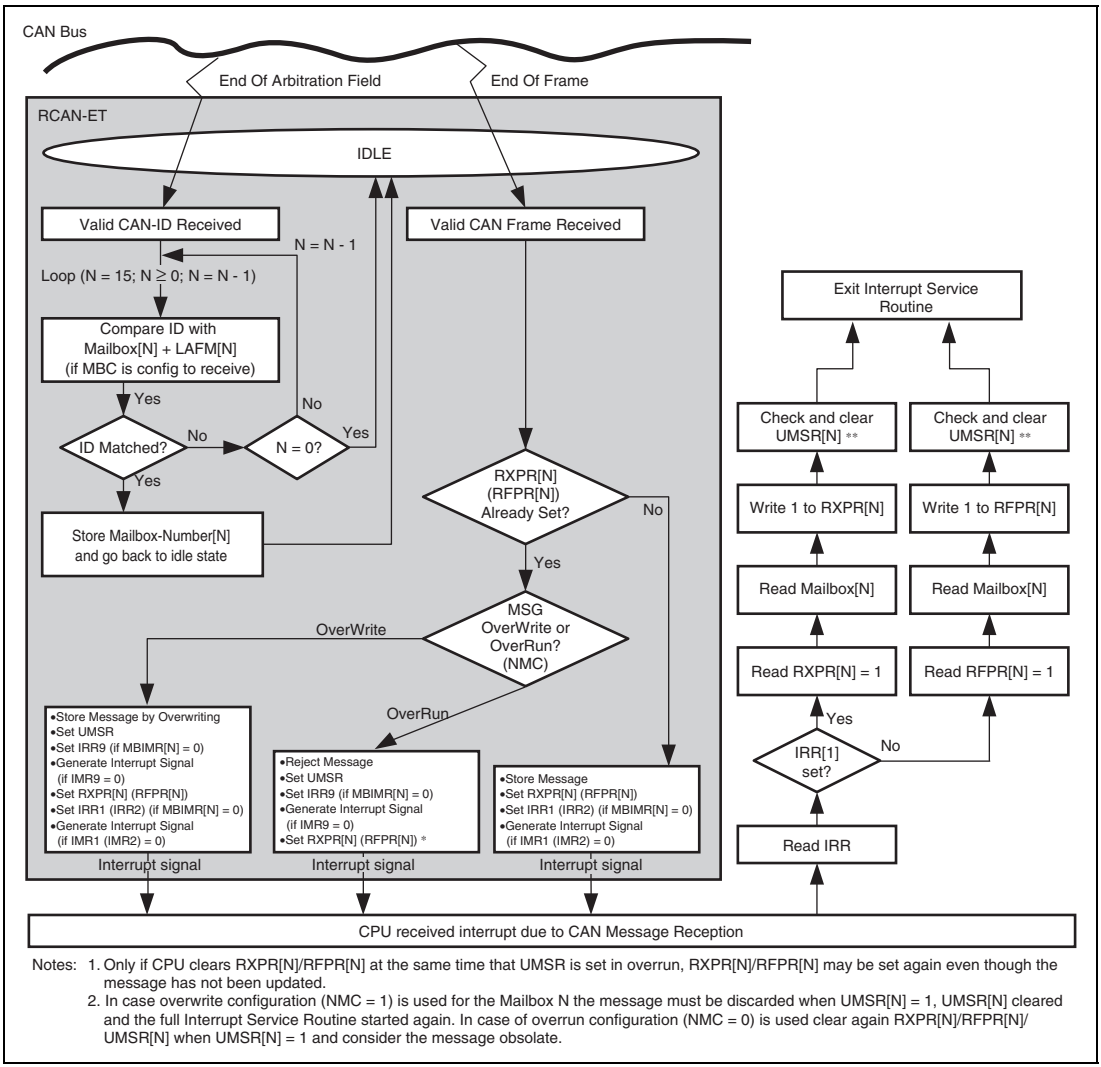
Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

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.

20.4.4 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 obsolete.

Figure 20.12 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.

20.4.5 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.

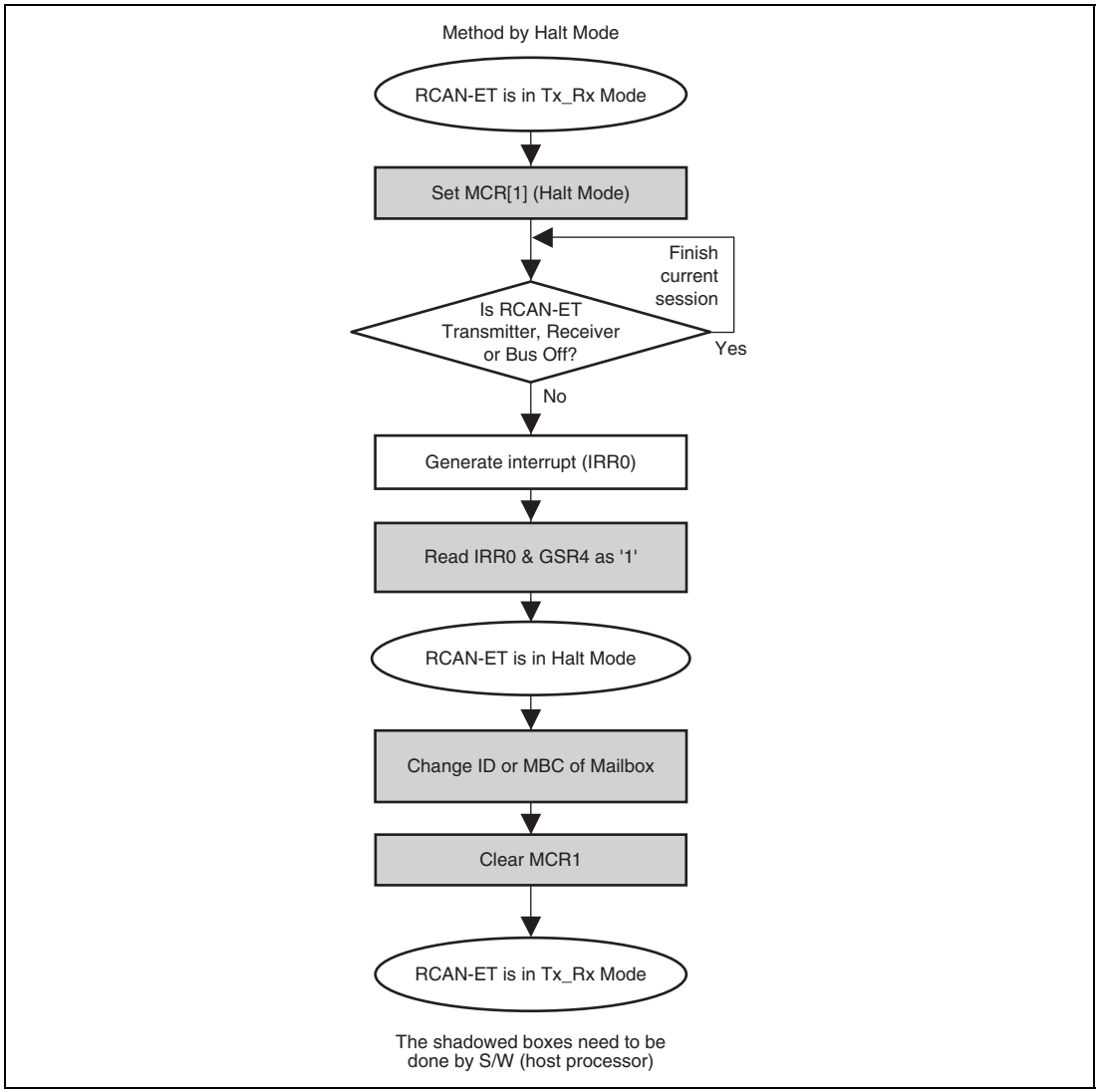


Figure 20.13 Change ID of Receive Box or Change Receive Box to Transmit Box

20.5 Interrupt Sources

Table 20.2 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 20.2 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.

20.6 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 20.14 shows a DTC transfer flowchart.

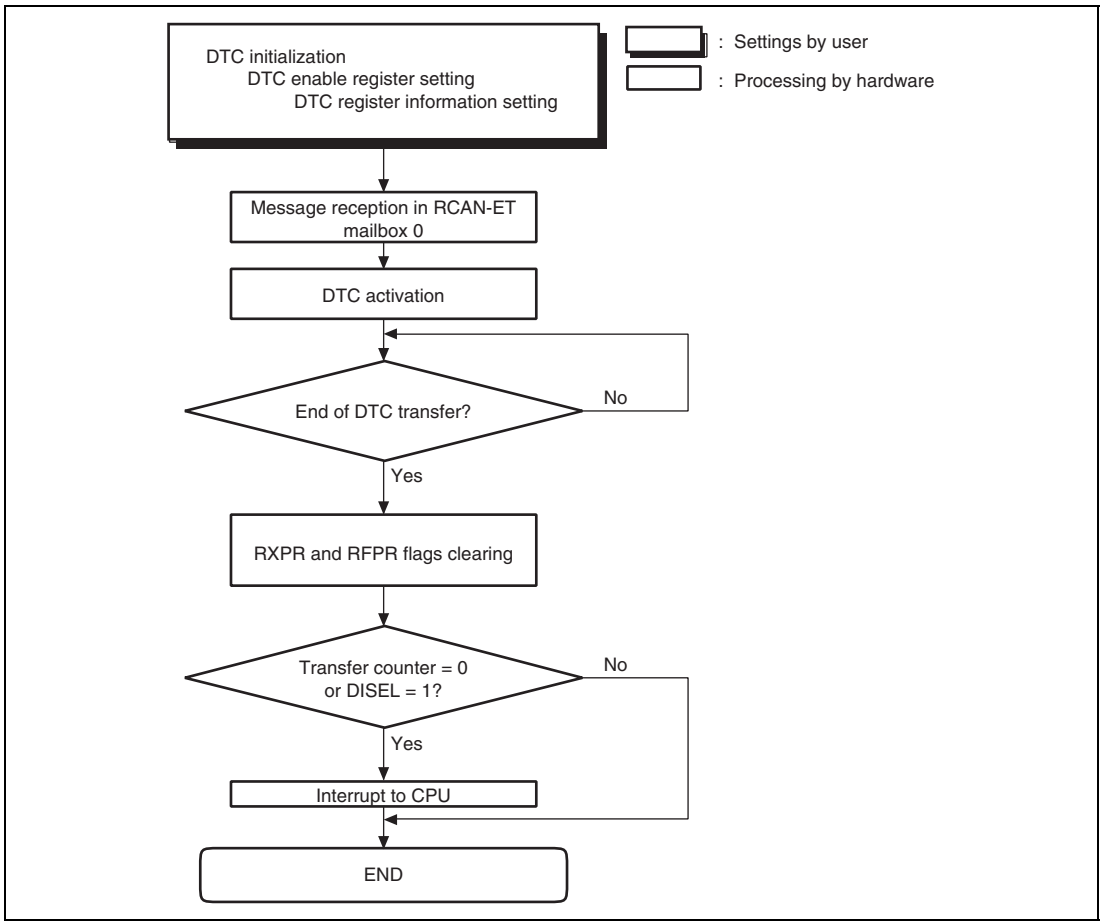
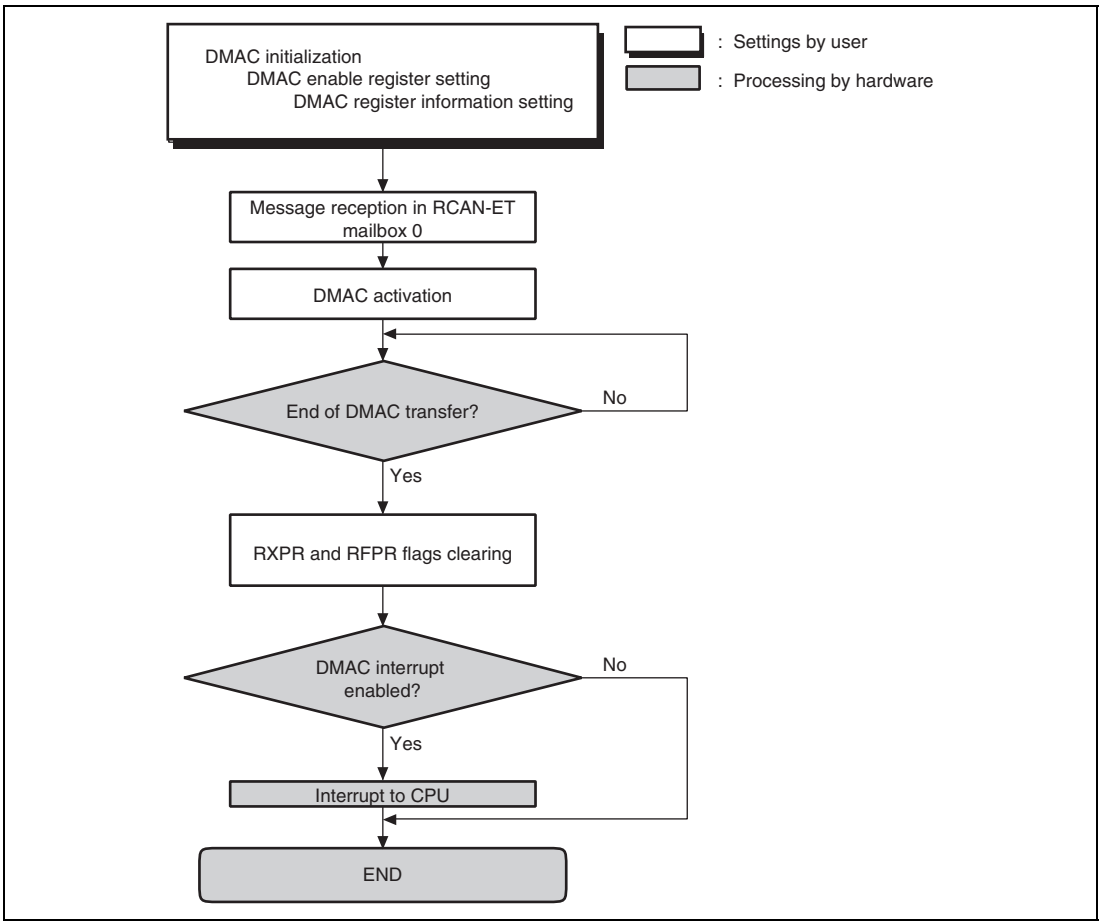


Figure 20.14 DTC Transfer Flowchart

20.7 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 20.15 shows a DMAC transfer flowchart.



20.15 DMAC Transfer Flowchart

20.8 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. Figure 20.16 shows a sample connection diagram.

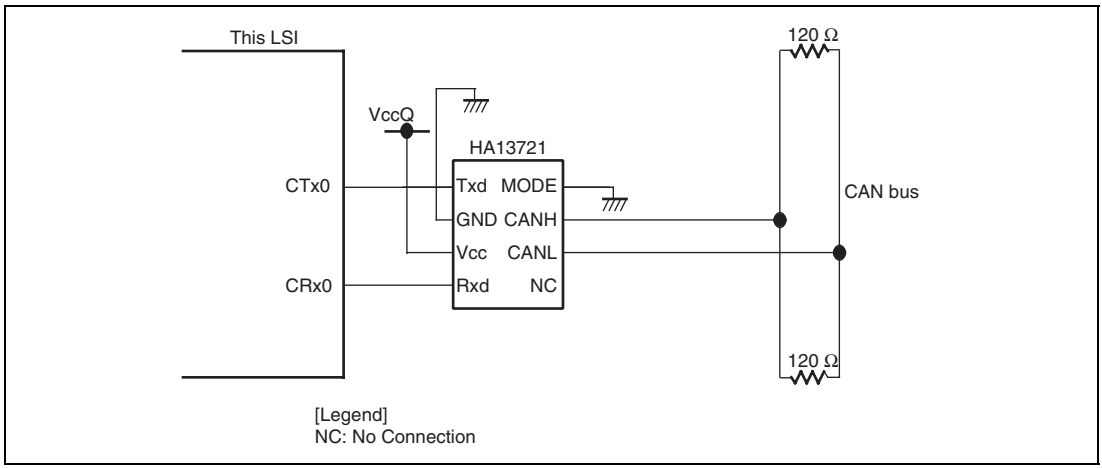


Figure 20.16 High-Speed CAN Interface Using HA13721

Section 21 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.

Table 21.1 Multiplexed Pins on Port A

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)
PA18 I/O (Port)	CK output* (BSC)	—	—	—	—	—	—
PA17 I/O (Port)	$\overline{RD}$ output* (BSC)	—	—	—	—	—	—
PA16 I/O (Port)	$\overline{WRL}$ output* (BSC)	—	—	—	—	—	—
PA15 I/O (Port)	$\overline{WRH}$ output* (BSC)	—	—	—	—	—	—
PA9 I/O (Port)	$\overline{CS3}$ output* (BSC)	—	IRQ3 input (INTC)	TCLKD input (MTU2)	SSLO I/O (RSPI)	SCK0 I/O (SCI)	—
PA8 I/O (Port)	$\overline{CS4}$ output* (BSC)	—	IRQ4 input (INTC)	TCLKC input (MTU2)	MISO I/O (RSPI)	RXD1 input (SCI)	—
PA7 I/O (Port)	$\overline{CS5}$ output* (BSC)	—	IRQ5 input (INTC)	TCLKB input (MTU2)	MOSI I/O (RSPI)	TXD1 output (SCI)	—
PA6 I/O (Port)	$\overline{CS6}$ output* (BSC)	—	IRQ6 input (INTC)	TCLKA input (MTU2)	RSPCK I/O (RSPI)	SCK1 I/O (SCI)	—
PA1 I/O (Port)	$\overline{CS1}$ output* (BSC)	—	IRQ5 input (INTC)	—	CTx0 output (RCAN-ET)	TXD0 output (SCI)	—
PA0 I/O (Port)	$\overline{CS0}$ output* (BSC)	—	IRQ4 input (INTC)	—	CRx0 input (RCAN-ET)	RXD0 input (SCI)	—

Note: * Can be used for the SH7239A and SH7237A only.

Table 21.2 Multiplexed Pins in Port B

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)
PB21 I/O (Port)	—	—	—	AUDATA1 output (AUD)	—	—	—
PB20 I/O (Port)	—	—	—	AUDATA0 output (AUD)	—	—	—
PB19 input (Port)	—	—	—	AUDATA3 output (AUD)	—	—	—
PB18 input (Port)	—	—	—	AUDATA2 output (AUD)	—	—	—
PB17 I/O (Port)	—	—	—	AUDCK output (AUD)	—	—	—
PB16 I/O (Port)	—	—	—	$\overline{\text{AUDSYNC}}$ output (AUD)	—	—	—
PB4 I/O (Port)	A20 output※1※6 (BSC)	$\overline{\text{BACK}}$ output※1※6 (BSC)	IRQ4 input※1 (INTC)	TIOC0D I/O※1 (MTU2)	$\overline{\text{WAIT}}$ input※1※6 (BSC)	SCK3 I/O※1 (SCIF)	$\overline{\text{BS}}$ output※1※6 (BSC)
PB3 I/O (Port)	A19 output※2※6 (BSC)	$\overline{\text{BREQ}}$ input※2※6 (BSC)	IRQ3 input※2 (INTC)	TIOC0C I/O※2 (MTU2)	—	TXD3 output※2 (SCIF)	$\overline{\text{AH}}$ output※2※6 (BSC)
PB2 I/O (Port)	A18 output※3※6 (BSC)	$\overline{\text{BACK}}$ output※3※6 (BSC) ※3	IRQ2 input※3 (INTC)	TIOC0B I/O※3 (MTU2)	—	RXD3 input※3 (SCIF)	—
PB1 I/O (Port)	A17 output※4※6 (BSC)	$\overline{\text{IRQOUT}}$ output※4 (INTC)	IRQ1 input※4 (INTC)	TIOC0A I/O※4 (MTU2)	—	—	$\overline{\text{ADTRG}}$ input※4 (ADC)
PB0 I/O (Port)	A16 output※5※6 (BSC)	—	IRQ0 input※5 (INTC)	TIOC2A I/O※5 (MTU2)	—	—	—

Notes: 1. When E10A is used ($\overline{\text{ASEMD0}}$ = L), the function is fixed to TCK input.
2. When E10A is used ($\overline{\text{ASEMD0}}$ = L), the function is fixed to TDO output.
3. When E10A is used ($\overline{\text{ASEMD0}}$ = L), the function is fixed to TDI input.
4. When E10A is used ($\overline{\text{ASEMD0}}$ = L), the function is fixed to $\overline{\text{TRST}}$ input.
5. When E10A is used ($\overline{\text{ASEMD0}}$ = L), the function is fixed to TMS input.
6. Available for the SH7239A and SH7237A only.

Table 21.3 Multiplexed Pins in Port C

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)
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)	SCK0 I/O (SCI)
PC8 I/O (Port)	A8 output* (BSC)	—	—	—	CRx0 input (RCAN-ET)	RXD0 input (SCI)	$\overline{\text{POE4}}$ input (POE2)
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)	—	—	—	—	—	$\overline{\text{ADTRG}}$ input (ADC)
PC0 I/O (Port)	A0 output* (BSC)	—	IRQ4 input (INTC)	—	$\overline{\text{POE0}}$ input (POE2)	—	—

Note: * Can be used for the SH7239A and SH7237A only.

Table 21.4 Multiplexed Pins in Port D

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)
PD15 I/O (Port)	D15 I/O* (BSC)	—	—	—	TIOC4DS I/O (MTU2S)	—	—
PD14 I/O (Port)	D14 I/O* (BSC)	—	—	—	TIOC4CS I/O (MTU2S)	—	—
PD13 I/O (Port)	D13 I/O* (BSC)	—	—	AUDCK output (AUD)	TIOC4BS I/O (MTU2S)	—	—
PD12 I/O (Port)	D12 I/O* (BSC)	—	—	$\overline{\text{AUDSYNC}}$ output (AUD)	TIOC4AS I/O (MTU2S)	—	—
PD11 I/O (Port)	D11 I/O* (BSC)	—	—	AUDATA3 output (AUD)	TIOC3DS I/O (MTU2S)	—	—
PD10 I/O (Port)	D10 I/O* (BSC)	—	—	AUDATA2 output (AUD)	TIOC3BS I/O (MTU2S)	—	—
PD9 I/O (Port)	D9 I/O* (BSC)	—	—	AUDATA1 output (AUD)	TIOC3CS I/O (MTU2S)	—	—
PD8 I/O (Port)	D8 I/O* (BSC)	—	—	AUDATA0 output (AUD)	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)	—	—	—	—	—	—

Note: * Can be used for the SH7239A and SH7237A only.

Table 21.5 Multiplexed Pins in Port E

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)
PE15 I/O (Port)	—	DACK1 output (DMAC)	$\overline{\text{IRQOUT}}$ output (INTC)	TIOC4D I/O (MTU2)	—	—	—
PE14 I/O (Port)	—	DACK0 output (DMAC)	—	TIOC4C I/O (MTU2)	—	—	—
PE13 I/O (Port)	—	—	$\overline{\text{MRES}}$ input (system control)	TIOC4B I/O (MTU2)	—	—	—
PE12 I/O (Port)	—	—	—	TIOC4A I/O (MTU2)	—	—	—
PE11 I/O (Port)	—	DACK3 output (DMAC)	—	TIOC3D I/O (MTU2)	—	—	—
PE10 I/O (Port)	—	DREQ3 input (DMAC)	—	TIOC3C I/O (MTU2)	SSL3 output (RSPI)	TXD2 output (SCI)	—
PE9 I/O (Port)	—	DACK2 output (DMAC)	—	TIOC3B I/O (MTU2)	—	—	—
PE8 I/O (Port)	—	DREQ2 input (DMAC)	—	TIOC3A I/O (MTU2)	SSL2 output (RSPI)	SCK2 I/O (SCI)	—
PE7 I/O (Port)	—	$\overline{\text{UBCTRG}}$ output (UBC)	—	TIOC2B I/O (MTU2)	SSL1 output (RSPI)	RXD2 input (SCI)	—
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)	—
PE4 I/O (Port)	—	—	IRQ4 input (INTC)	TIOC1A I/O (MTU2)	$\overline{\text{POE8}}$ input (POE2)	SCK3 I/O (SCIF)	—
PE3 I/O (Port)	—	TEND1 output (DMAC)	—	TIOC0D I/O (MTU2)	TIOC4DS I/O (MTU2S)	—	—
PE2 I/O (Port)	—	DREQ1 input (DMAC)	—	TIOC0C I/O (MTU2)	TIOC4CS I/O (MTU2S)	—	—
PE1 I/O (Port)	—	TEND0 output (DMAC)	—	TIOC0B I/O (MTU2)	TIOC4BS I/O (MTU2S)	—	—
PE0 I/O (Port)	—	DREQ0 input (DMAC)	—	TIOC0A I/O (MTU2)	TIOC4AS I/O (MTU2S)	—	—

Table 21.6 Multiplexed Pins in Port F

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)
PF15 input (Port)	AN15 input (ADC)	—	—	—	—	—	—
PF14 input (Port)	AN14 input (ADC)	—	—	—	—	—	—
PF13 input (Port)	AN13 input (ADC)	—	—	—	—	—	—
PF12 input (Port)	AN12 input (ADC)	—	—	—	—	—	—
PF11 input (Port)	AN11 input (ADC)	—	—	—	—	—	—
PF10 input (Port)	AN10 input (ADC)	—	—	—	—	—	—
PF9 input (Port)	AN9 input (ADC)	—	—	—	—	—	—
PF8 input (Port)	AN8 input (ADC)	—	—	—	—	—	—
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.

21.1 Register Descriptions

The PFC has the following registers. See section 28, List of Registers for register addresses and register states in each operating mode.

Table 21.7 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 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 H	PBIORH	R/W	H'0000	H'FFFE3884	8, 16, 32
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE3886	8, 16
Port B control register H2	PBPCRH2	R/W	H'0000	H'FFFE388C	8, 16, 32
Port B control register H1	PBPCRH1	R/W	H'0000	H'FFFE388E	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 H	PBPCRH	R/W	H'0000	H'FFFE38A8	8, 16, 32
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 L	PDIORL	R/W	H'0000	H'FFFE3986	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

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D control register L1	PDCRL1	R/W	H'0000	H'FFFE3996	8, 16
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
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

21.1.1 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. PAIORH and PAIORL are enabled when the port A pins are functioning as general-purpose inputs/outputs. 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 3 of PAIORH and bits 14 to 10 and bits 5 to 2 of PAIORL are reserved. These bits are always read as 0. The write value should always be 0.

(1) Port A I/O Register H (PAIORH)

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

(2) 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	-	-	-	-	-	PA9 IOR	PA8 IOR	PA7 IOR	PA6 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	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

21.1.2 Port A Control Registers H1 and L1 to L4 (PACRH1 and PACRL1 to PACRL4)

PACRH1 and PACRL1 to PACRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port A.

(1) Port A Control Register H1 (PACRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PA18MD[2:0]			-	PA17MD[2:0]			-	PA16MD[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/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 when the MCU operating mode is mode 2 or mode 4 (user program mode).

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA18MD[2:0]	000*[1]	R/W	PA18 Mode 000: PA18 I/O (port) 001: CK output (BSC)*[2] 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	R/W	PA17 Mode 000: PA17 I/O (port) 001: $\overline{RD}$ output (BSC)*2 010: 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	R/W	PA16 Mode 000: PA16 I/O (port) 001: $\overline{WRL}$ output (BSC)*2 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Notes: 1. The initial value is 001 when the MCU operating mode is mode 2 or mode 4 (user program mode).
2. Can be used for SH7239A and SH7237A only.

(2) 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]			-	-	-	-	-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R	R	R	R

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	R/W	PA15 Mode 000: PA15 I/O (port) 001: $\overline{\text{WRH}}$ output (BSC)* 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11 to 0	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.

Note: * Can be used for SH7239A and SH7237A only.

(3) Port A Control Register L3 (PACRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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	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 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 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: Setting prohibited
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 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: Setting prohibited

Note: * Can be used for SH7239A and SH7237A only.

(4) 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]			-	-	-	-	-	-	-	-
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	R	R	R	R	R	R

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 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: Setting prohibited
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 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: Setting prohibited
7 to 0	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.

Note: * Can be used e for SH7239A and SH7237A only.

(5) Port A Control Register L1 (PACRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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	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 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 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: Setting prohibited
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 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: Setting prohibited

Note: * Can be used for SH7239A and SH7237A only.

21.1.3 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.

(1) 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
	-	-	-	-	-	-	-	-	-	-	-	-	-	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	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PA18PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
1	PA17PCR	0	R/W	
0	PA16PCR	0	R/W	

(2) 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	-	-	-	-	-	PA9 PCR	PA8 PCR	PA7 PCR	PA6 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	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PA9PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
8	PA8PCR	0	R/W	
7	PA7PCR	0	R/W	
6	PA6PCR	0	R/W	
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PA1PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
0	PA0PCR	0	R/W	

21.1.4 Port B I/O Register H, L (PBIORH, PBIORL

PBIORH and PBIORL are 16-bit readable/writable registers that are used to set the pins on port B as inputs or outputs. PBIORH and PBIORL are enabled when the port B pins are functioning as general-purpose inputs/outputs and TIOC inputs/outputs in MTU2. In other states, PBIORL is disabled. A given pin on port B will be an output pin if the corresponding bit in PBIORH or PBIORL is set to 1, and an input pin if the bit is cleared to 0.

(1) Port B I/O Register H (PBIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB21 IOR	PB20 IOR	PB19 IOR	PB18 IOR	PB17 IOR	PB16 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	R/W	R/W	R/W	R/W	R/W

(2) Port B I/O Register L (PBIORL))

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

21.1.5 Port B Control Register H1, H2, L1, and L2 (PBCRH1, PBCRH2), PBCRL1, and PBCRL2)

PBCRH1, PBCRH2, PBCRL1, and PBCRL2 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port B.

(1) Port B Control Register H2 (PBCRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PB21MD[2:0]			-	PB20MD[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 This bit is read as 0. The write value should always be 0.
6 to 4	PB21MD[2:0]	000	R/W	PB21 Mode 000: PB21 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: AUDATA1 output (AUD) 101: Setting prohibited) 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PB20MD[2:0]	000	R/W	PB20 Mode 000: PB20 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: AUDATA0 output (AUD) 101: Setting prohibited) 110: Setting prohibited 111: Setting prohibited

(2) Port B Control Register H1 (PBCRH1))

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB19MD[2:0]			-	PB18MD[2:0]			-	PB17MD[2:0]			-	PB16MD[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	PB19MD[2:0]	000	R/W	PB19 Mode 000: PB19 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: AUDATA3 output (AUD) 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	PB18MD[2:0]	000	R/W	PB18 Mode 000: PB18 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: AUDATA2 output (AUD) 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	PB17MD[2:0]	000	R/W	PB17 Mode 000: PB17 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: AUDCK output (AUD) 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	PB16MD[2:0]	000	R/W	PB16 Mode 000: PB16 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: Setting prohibited 100: $\overline{\text{AUDSYNC}}$ output (AUD) 101: Setting prohibited) 110: Setting prohibited 111: Setting prohibited

(3) Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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	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
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	PB4MD[2:0]	000	R/W	PB4 Mode 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: $\overline{\text{WAIT}}$ input (BSC)* 110: SCK3 I/O (SCIF3) 111: $\overline{\text{BS}}$ input (BSC)*

Note: * Can be used for SH7239A and SH7237A only.

(4) 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

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	R/W	PB3 Mode 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: Setting prohibited 110: TXD3 output (SCIF3) 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	R/W	PB2 Mode 000: PB2 I/O (port) 001: A18 output (BSC)* 010: $\overline{\text{BACK}}$ input (BSC)* 011: IRQ2 input (INTC) 100: TIOC0B I/O (MTU2) 101: Setting prohibited 110: RXD3 output (SCIF3) 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	PB1MD[2:0]	000	R/W	PB1 Mode 000: PB1 I/O (port) 001: A17 output (BSC)* 010: $\overline{\text{IRQOUT}}$ input (INTC) 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	R/W	PB0 Mode 000: PB0 I/O (port) 001: A16 output (BSC)* 010: Setting prohibited 011: IRQ0 input (INTC) 100: TIOC2A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * Can be used for SH7239A and SH7237A only.

21.1.6 Port B Pull-Up MOS Control Registers H and L (PBPCRH, PBPCRL)

PBPCRH and PBPCRL control on/off of the input pull-up MOS of port B in bits.

(1) Port B Pull-Up MOS Control Register H (PBPCRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB21 PCR	PB20 PCR	PB19 PCR	PB18 PCR	PB17 PCR	PB16 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	PB21PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
4	PB20PCR	0	R/W	
3	PB19PCR	0	R/W	
2	PB18PCR	0	R/W	
1	PB17PCR	0	R/W	
0	PB16PCR	0	R/W	

(2) 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
	-	-	-	-	-	-	-	-	-	-	-	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
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 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB4PCR	0	R/W	The corresponding input pull-up MOS turns on when one of these bits is set to 1.
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

21.1.7 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. PCIORL is enabled when the port C pins are functioning as general-purpose inputs/outputs and TIOC inputs/outputs in MTU2. 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.

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

21.1.8 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.

(1) 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

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	PC15MD[2:0]	000	R/W	PC15 Mode 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

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	PC14MD[2:0]	000	R/W	PC14 Mode 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.
6 to 4	PC13MD[2:0]	000	R/W	PC13 Mode 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.

Bit	Bit Name	Initial Value	R/W	Description
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: * Can be used for SH7239A and SH7237A only.

(2) 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

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 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 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
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	PC9MD[2:0]	000	R/W	PC9 Mode 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: SCK0 I/O (SCI)
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 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: $\overline{\text{POE4}}$ input (POE2)

Note: * Can be used for SH7239A and SH7237A only.

(3) 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

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: * Can be used for SH7239A and SH7237A only.

(4) 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

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 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 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
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	PC1MD[2:0]	000	R/W	PC1 Mode 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: $\overline{\text{ADTRG}}$ input (ADC)
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PC0MD[2:0]	000	R/W	PC0 Mode 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: * Can be used for SH7239A and SH7237A only.

21.1.9 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	

21.1.10 Port D I/O Register L (PDIORL)

PDIORL is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. PDIORL is enabled when the port D pins are functioning as general-purpose inputs/outputs and TIOC inputs/outputs in MTU2S. In other states, PDIORL is disabled. A given pin on port D will be an output pin if the corresponding bit in PDIORL is set to 1, and an input pin if the bit is cleared to 0.

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

21.1.11 Port D Control Registers L1 to L4 (PDCRL1 to PDCRL4)

PDCRL1 to PDCRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port D.

(1) 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

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 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.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PD14MD[2:0]	000	R/W	PD14 Mode 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
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 000: PD13 I/O (port) 001: D13 I/O (BSC)* 010: Setting prohibited 011: Setting prohibited 100: AUDCK output (AUD) 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 000: PD12 I/O (port) 001: D12 I/O (BSC)* 010: Setting prohibited 011: Setting prohibited 100: $\overline{\text{AUDSYNC}}$ output (AUD) 101: TIOC4AS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Note: * Can be used for SH7239A and SH7237A only.

(2) 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

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 000: PD11 I/O (port) 001: D11 I/O (BSC)* 010: Setting prohibited 011: Setting prohibited 100: AUDATA3 output (AUD) 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 000: PD10 I/O (port) 001: D10 I/O (BSC)* 010: Setting prohibited 011: Setting prohibited 100: AUDATA2 output (AUD) 101: TIOC3BS I/O (MTU2S) 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	PD9MD[2:0]	000	R/W	PD9 Mode 000: PD9 I/O (port) 001: D9 I/O (BSC)* 010: Setting prohibited 011: Setting prohibited 100: AUDATA1 output (AUD) 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 000: PD8 I/O (port) 001: D8 I/O (BSC)* 010: Setting prohibited 011: Setting prohibited 100: AUDATA0 output (AUD) 101: TIOC3AS I/O (MTU2S) 110: Setting prohibited 111: Setting prohibited

Note: * Can be used for SH7239A and SH7237A only.

(3) 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

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 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 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
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	PD5MD[2:0]	000	R/W	PD5 Mode 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 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: * Can be used for SH7239A and SH7237A only.

(4) 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

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 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 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
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	PD1MD[2:0]	000	R/W	PD1 Mode 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 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: * Can be used for SH7239A and SH7237A only.

21.1.12 Port D Pull-Up MOS Control Register L (PDPCRL))

PDPCRL controls on/off of the input pull-up MOS of port D in bits.

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	

21.1.13 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. PEIORL is enabled when the port E pins are functioning as general-purpose inputs/outputs 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.

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

21.1.14 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.

(1) 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 000: PE15 I/O (port) 001: Setting prohibited 010: DACK1 output (DMAC) 011: $\overline{\text{IRQOUT}}$ output (INTC) 100: TIOC4D I/O (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.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PE14MD[2:0]	000	R/W	PE14 Mode 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: Setting prohibited
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 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: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PE12MD[2:0]	000	R/W	PE12 Mode 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: Setting prohibited

(2) 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 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: Setting prohibited
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 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: 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	PE9MD[2:0]	000	R/W	PE9 Mode 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: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PE8MD[2:0]	000	R/W	PE8 Mode 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: Setting prohibited

(3) 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.

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PE7MD[2:0]	000	R/W	PE7 Mode 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: Setting prohibited
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 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 (SCIF3) 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 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 (SCIF3) 111: Setting prohibited

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 to 0	PE4MD[2:0]	000	R/W	PE4 Mode 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 (SCIF3) 111: Setting prohibited

(4) 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 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: Setting prohibited

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 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: Setting prohibited
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 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: 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	PE0MD[2:0]	000	R/W	PE0 Mode 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: Setting prohibited

21.1.15 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	

21.1.16 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, 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 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	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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
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
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

Bit	Bit Name	Initial Value	R/W	Description
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.

21.1.17 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 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 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 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 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.

21.2 Pull-Up MOS Control by Pin Function

Table 21.8 shows the pull-up MOS control by pin function and the pull-up MOS control in each operating mode.

Table 21.8 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	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 IRQ6 input (INTC)							
$\overline{\text{MRES}}$ input (System control)							
$\overline{\text{POE0}}$, $\overline{\text{POE4}}$, and $\overline{\text{POE8}}$ input (POE2)							
RXD0 to RXD3 input (SCI, SCIF)							
SCK0 to SCK3 input (SCI, SCIF)							
CRx0 input (RCAN-ET)							
$\overline{\text{ADTRG}}$ input (ADC)							
SSLO and RSPCR input (RSPI)							
MISO and MOSI input (RSPI)							

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)							
$\overline{WRH}$ and $\overline{WRL}$ output (BSC)							
$\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$ to $\overline{CS6}$ output (BSC)							
$\overline{BS}$ output, $\overline{AH}$ output, and $\overline{BACK}$ output (BSC)							
DACK0 to DACK3 output (DMAC)							
TEND0 output and TEND1 output (DMAC)							
$\overline{IRQOUT}$ output (INTC)							
$\overline{UBCTRG}$ output (UBC)							
TXD0 to TXD3 output (SCI, SCIF)							
SCK0 to SCK3 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)							

Pin Function	Power-On Reset	Manual Reset	Software Standby	Sleep	When Oscillation Stop is Detected	When POE Function is Used	Normal Operation
Data bus input/output (BSC)	Off	Off	Off	Off	Off	Off	Off
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)							

[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), 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.

21.3 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 21.9 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 21.9 Transmission Format of Input Function Allocated on Multiple Pins

OR Type	AND Type
TCLKA, TCLKB, TCLKC, TCLKD (MTU2)	IRQ0 to IRQ5 (INTC)
TIOC0A, TIOC0B, TIOC0C, TIOC0D (MTU2)	$\overline{\text{ADTRG}}$ (ADC)
TIOC1A, TIOC1B, TIOC2A (MTU2)	CRx0 (RCAN-ET)
TIC5U, TIC5V, TIC5W (MTU2)	
TIOC3AS, TIOC3BS, TIOC3CS, TIOC3DS (MTU2S)	
TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS (MTU2S)	
SCK0, SCK2, SCK3, RXD0, RXD2, RXD3 (SCI, SCIF)	

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 IRQ edge for the pins that are multiplexed with I/O and IRQ, the corresponding edge is detected.
3. Do not set functions other than settable functions. Otherwise, correct operation cannot be guaranteed.

Section 22 I/O Ports

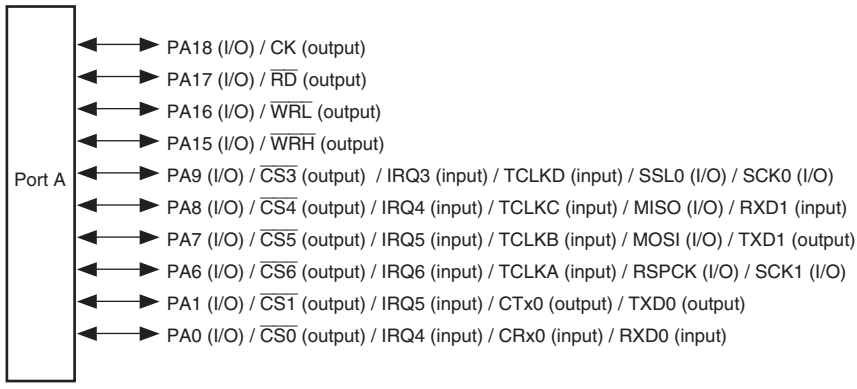
The I/O ports are comprised of six ports: A, B, C, D, E, and F. Port A is a 10-pin, port B is an 11-pin, port C is a 16-pin, port D is a 16-pin, and port E is a 16-pin I/O ports. Port F is a 16-pin input-only port.

All the port pins are multiplexed with other pin functions. The functions of the multiplex pins are selected by means of the pin function controller (PFC). For details, see section 21, Pin Function Controller (PFC).

Each port is provided with data registers for storing the pin data.

22.1 Port A

Port A is an I/O port with 10 pins shown in figure 22.1.



Note: CS0, CS1, CS3 to CS6, WRH, WRL, RD, and CK can be used for the SH7239A and SH7237A only.

Figure 22.1 Port A

22.1.1 Register Descriptions

Port A has the following registers. See section 28, List of Registers for details on the register address and states in each operating mode.

Table 22.1 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

22.1.2 Port A Data Registers H and L (PADRH and PADRL)

PADRH and PADRL are 16-bit readable/writable registers that store port A data. 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 22.2 summarizes read/write operations of port A data registers.

(1) Port A Data Register H (PADRH)

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	PA18DR	0	R/W	See table 22.2.
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

(2) 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	-	-	-	-	-	PA9 DR	PA8 DR	PA7 DR	PA6 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	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 22.2.
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PA9DR	0	R/W	See table 22.2.
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PA1DR	0	R/W	See table 22.2.
0	PA0DR	0	R/W	

Table 22.2 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.

22.1.3 Port A Port Registers H and L (PAPRH and PAPRL)

PAPRH and PAPRL are 16-bit read-only registers, which always return the states of the pins regardless of the PFC setting. However, when PA9 pin is specified as the RSPI function, the states of the corresponding pins cannot be read.

(1) Port A Port Register H (PAPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	PA18PR	PA17PR	PA16PR
Initial value:	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 3	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
2	PA18PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
1	PA17PR	Pin state	R	
0	PA16PR	Pin state	R	

(2) 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	-	-	-	-	-	PA9 PR	PA8 PR	PA7 PR	PA6 PR	-	-	-	-	PA1 PR	PA0 PR
Initial value:	*	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	PA15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PA9PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified. However, when PA9 pin is specified as the RSPI function, the states of the corresponding pins cannot be read.
8	PA8PR	Pin state	R	
7	PA7PR	Pin state	R	
6	PA6PR	Pin state	R	
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	PA1PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
0	PA0PR	Pin state	R	

22.2 Port B

Port B is an I/O port with 11 pins shown in figure 22.2.

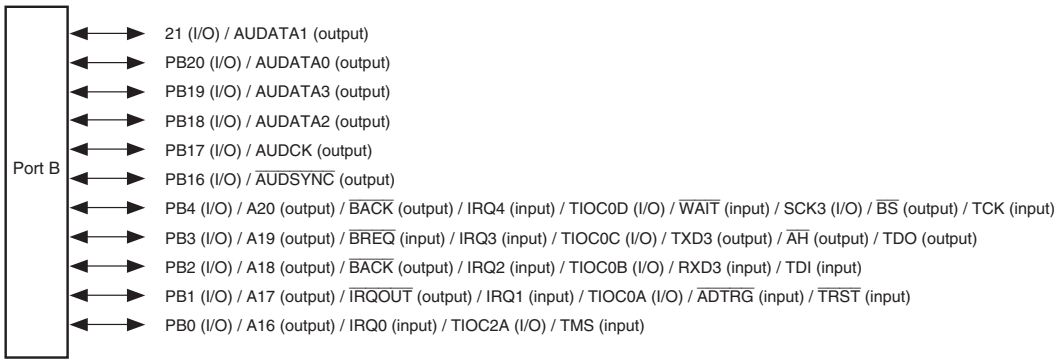


Figure 22.2 Port B

22.2.1 Register Descriptions

Port B has the following registers. See section 28, List of Registers for details on the register address and states in each operating mode.

Table 22.3 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register H	PBDRH	R/W	H'0000	H'FFFE3880	8, 16, 32
Port B data register L	PBDRL	R/W	H'0000	H'FFFE3882	8, 16
Port B port register H	PBPRL	R	—	H'FFFE389C	8, 16, 32
Port B port register L	PBPRL	R	—	H'FFFE389E	8, 16

22.2.2 Port B Data Registers H and L (PBDRH and PBDRL)

PBDRH and PBDRL are 16-bit readable/writable registers that store port B data. When a pin function is general output, if a value is written to PBDRH or PBDRL, the value is output directly from the pin, and if PBDRH or PBDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PBDRH or PBDRL is read, the pin state, not the register value, is returned directly. If a value is written to PBDRH or PBDRL, although that value is written into PBDRH or PBDRL, it does not affect the pin state.

Table 22.4 summarizes read/write operations of port B data registers.

(1) Port B Data Register H (PBDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR
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	PB21DR	0	R/W	See table 22.4.
4	PB20DR	0	R/W	
3	PB19DR	0	R/W	
2	PB18DR	0	R/W	
1	PB17DR	0	R/W	
0	PB16DR	0	R/W	

(2) Port B Data Register L (PBDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
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	Bit Name	Initial Value	R/W	Description
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB4DR	0	R/W	See table 22.4.
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 22.4 Port B Data Registers H and L (PBDRH and PBDRL) Read/Write Operations

PBIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PBDRH and PBDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PBDRH and PBDRL, but it has no effect on pin state.
1	General output	PBDRH or PBDRL value	The value written is output from the pin.
	Other than general output	PBDRH or PBDRL value	Can write to PBDRH and PBDRL, but it has no effect on pin state.

22.2.3 Port B Port Registers H and L (PBPRH and PBPRL)

PBPRH and PBPRL are 16-bit read-only registers, which return the states of the pins. However, note that the pin state cannot be read when PB3 is set to the SCIF function and the TE bit in SCSCR and SPB2IO bit in SCSPTR are 0.

(1) Port B Port Register H (PBPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB21 PR	PB20 PR	PB19 PR	PB18 PR	PB17 PR	PB16 PR
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. The write value should always be 0.
5	PB21PR	Pin state	R	The pin state is returned. These bits cannot be modified.
4	PB20PR	Pin state	R	
3	PB19PR	Pin state	R	
2	PB18PR	Pin state	R	
1	PB17PR	Pin state	R	
0	PB16PR	Pin state	R	

(2) Port B Port Register L (PBPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	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 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB4PR	Pin state	R	The pin state is returned. Note that the pin state cannot be read when PB3 is set to the SCIF function and the TE bit in SCSCR_3 and SPB2IO bit in SCSPTR_3 are 0.
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	PB0PR	Pin state	R	

22.3 Port C

Port C is an I/O port with 16 pins shown in figure 22.3.

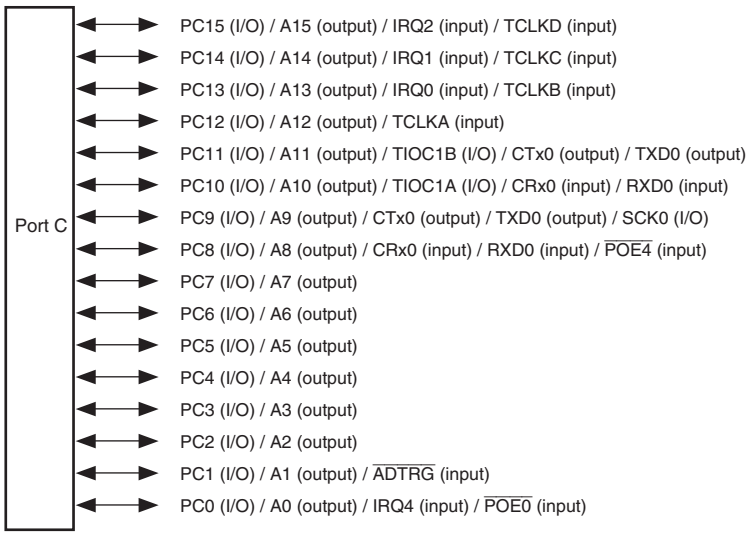


Figure 22.3 Port C

22.3.1 Register Descriptions

Port C has the following registers. See section 28, List of Registers for details on the register address and states in each operating mode.

Table 22.5 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

22.3.2 Port C Data Register L (PCDRL)

PCDRL is a 16-bit readable/writable register that store port C data. 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 22.6 summarizes read/write operations of port C data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 DR	PC14 DR	PC13 DR	PC12 DR	PC11 DR	PC10 DR	PC9 DR	PC8 DR	PC7 DR	PC6 DR	PC5 DR	PC4 DR	PC3 DR	PC2 DR	PC1 DR	PC0 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	PC15DR	0	R/W	See table 22.6.
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	
8	PC8DR	0	R/W	
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 22.6 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.

22.3.3 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.

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	

22.4 Port D

Port D is an I/O port with 16 pins shown in figure 22.4.

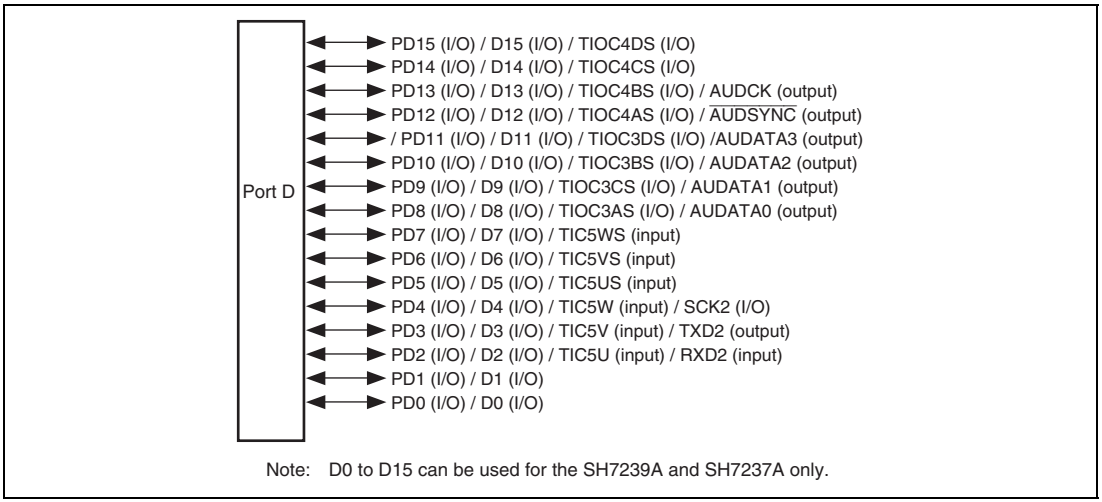


Figure 22.4 Port D

22.4.1 Register Descriptions

Port D has the following registers. See section 28, List of Registers for details on the register address and states in each operating mode.

Table 22.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register L	PDDRL	R/W	H'0000	H'FFFE3982	8, 16
Port D port register L	PDPRL	R	—	H'FFFE399E	8, 16

22.4.2 Port D Data Register L (PDDRL)

PDDRL is a 16-bit readable/writable registers that stores port D data. When a pin function is general output, if a value is written to PDDRL, the value is output directly from the pin, and if PDDRL is read, the register value is returned directly regardless of the pin state. When a pin function is general input, if PDDRL is read, the pin state, not the register value, is returned directly. If a value is written to PDDRL, although that value is written into PDDRL, it does not affect the pin state.

Table 22.8 summarizes read/write operations of port D data register.

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 22.8.
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 22.8 Port D Data Register L (PDDRL) Read/Write Operations

PDIORL	Pin Function	Read	Write
0	General input	Pin state	Can write to PDDRL, but it has no effect on pin state.
	Other than general input	Pin state	Can write to PDDRL, but it has no effect on pin state.
1	General output	PDDRL value	The value written is output from the pin.
	Other than general output	PDDRL value	Can write to PDDRL, but it has no effect on pin state.

22.4.3 Port D Port Register L (PDPRL)

PDPRL is a 16-bit read-only register, which always returns the states of the pins regardless of the PFC setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 PR	PD14 PR	PD13 PR	PD12 PR	PD11 PR	PD10 PR	PD9 PR	PD8 PR	PD7 PR	PD6 PR	PD5 PR	PD4 PR	PD3 PR	PD2 PR	PD1 PR	PD0 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	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	
1	PD1PR	Pin state	R	
0	PD0PR	Pin state	R	

22.5 Port E

Port D is an I/O port with 16 pins shown in figure 22.5

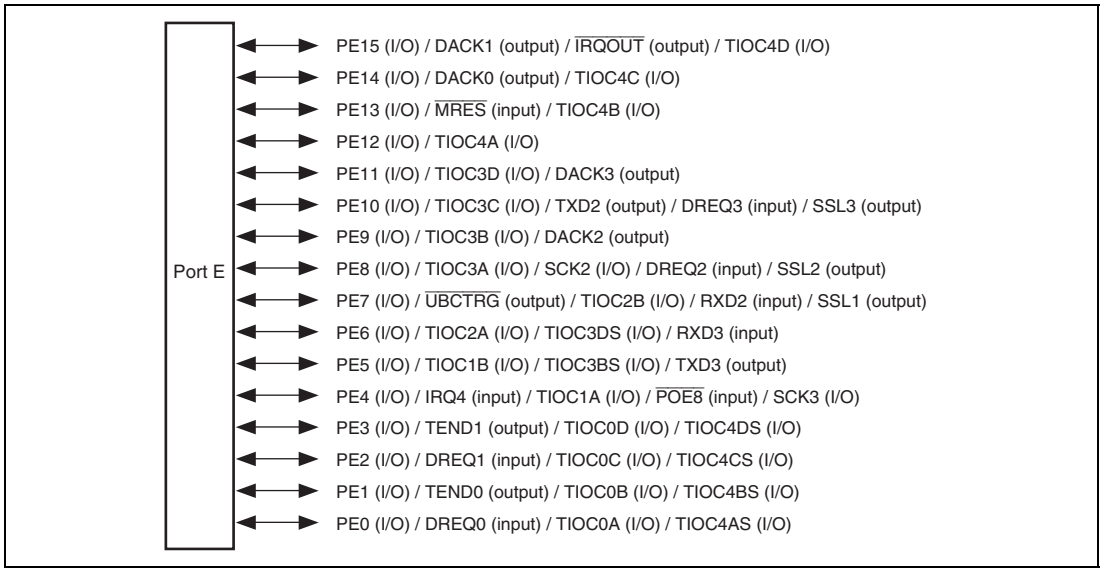


Figure 22.5 Port E

22.5.1 Register Descriptions

Port E has the following registers. See section 28, List of Registers for details on the register address and states in each operating mode.

Table 22.9 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

22.5.2 Port E Data Register L (PEDRL)

PEDRL is a 16-bit readable/writable register that stores port E data. 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 22.10 summarizes read/write operations of port E data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9 DR	PE8 DR	PE7 DR	PE6 DR	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 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	PE15DR	0	R/W	See table 22.10.
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 22.10 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.

22.5.3 Port E Port Register L (PEPRL)

PEPRL is a 16-bit read-only register, which returns the states of the pins. However, note that the pin state cannot be read when PE10, PE8, and PE7 are set to the RSPI function and when PE5 is set to the SCIF function and the TE bit in SCSCR_3 and SPB2IO bit in SCSPTR_3 are 0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 PR	PE14 PR	PE13 PR	PE12 PR	PE11 PR	PE10 PR	PE9 PR	PE8 PR	PE7 PR	PE6 PR	PE5 PR	PE4 PR	PE3 PR	PE2 PR	PE1 PR	PE0 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	PE15PR	Pin state	R	The pin state is returned. Note that the pin state cannot be read when PE5 is set to the SCIF function and the TE bit in SCSCR_3 and SPB2IO bit in SCSPTR_3 are 0.
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	
5	PE5PR	Pin state	R	
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	

22.6 Port F

Port F is an I/O port with 16 pins shown in figure 22.6.

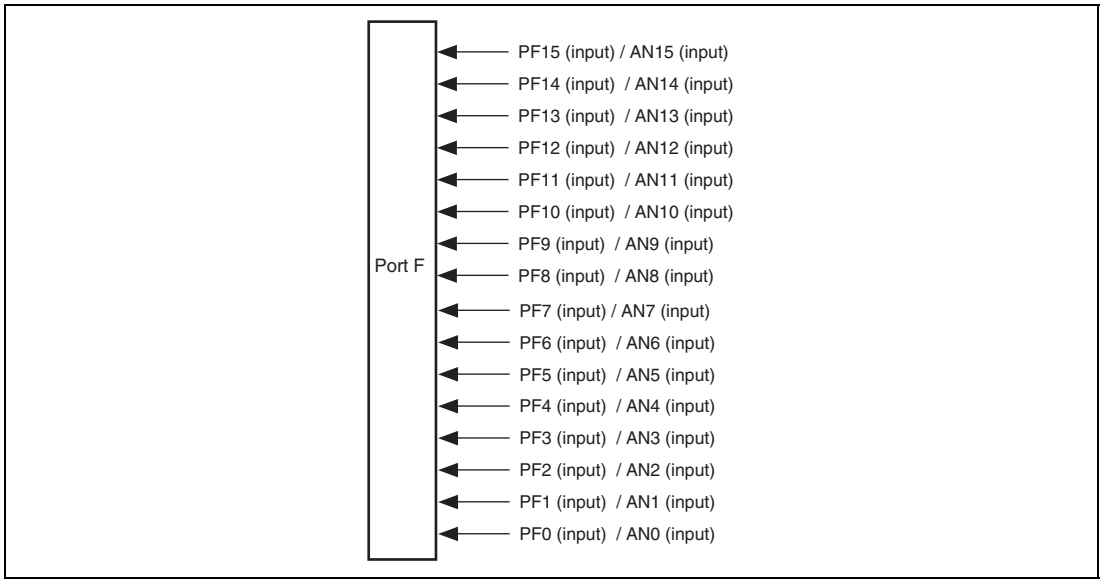


Figure 22.6 Port F

22.6.1 Register Descriptions

Port F has the following registers. See section 28, List of Registers for details on the register address and states in each operating mode.

Table 22.11 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register L	PFDRL	R	—	H'FFFE3A82	8, 16

22.6.2 Port F Data Register L (PFDRL)

PFDRL is a 16-bit read-only register that stores port F data.

Even if a value is written to a bit, the pin state is not affected. If a PFDRL bit 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 22.12 summarizes read/write operations of port F data register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
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	PF15DR	Pin state	R	See table 22.12.
14	PF14DR	Pin state	R	
13	PF13DR	Pin state	R	
12	PF12DR	Pin state	R	
11	PF11DR	Pin state	R	
10	PF10DR	Pin state	R	
9	PF9DR	Pin state	R	
8	PF8DR	Pin state	R	
7	PF7DR	Pin state	R	
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 22.12 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)

[Legend]

n: 0 to 15

22.7 Usage Notes

22.7.1 Handling of Unused pins

Levels on unused pins of port F should be fixed by connection to AVCC or AVSS via resistances. For handling of the NMI, EXTAL, XTAL, $\overline{\text{WDTOVF}}$, $\overline{\text{TRST}}$, TMS, TCK, TDO, and TDI pins, follow the instructions in the sections on the corresponding modules.

Other unused pins should be connected to VCC or GND via resistors to fix high or low levels on the pins.

Section 23 Flash Memory (ROM)

The SH7239 and SH7237 Groups incorporate 512 Kbytes or 256 Kbytes of flash memory (ROM). For the capacity of flash memory (ROM) in each product, see section 1.2, List of Products.

23.1 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'0007FFFF, undefined data is read and programming and erasing are ignored when the user boot MAT is selected.

  User MAT: 512 Kbytes or 256 Kbytes

  User boot MAT: 32 Kbytes

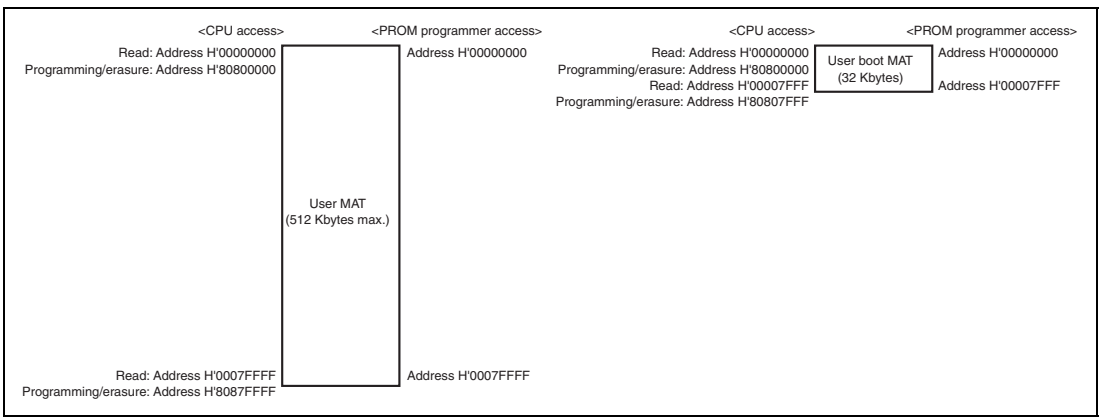


Figure 23.1 Memory MAT Configuration in ROM

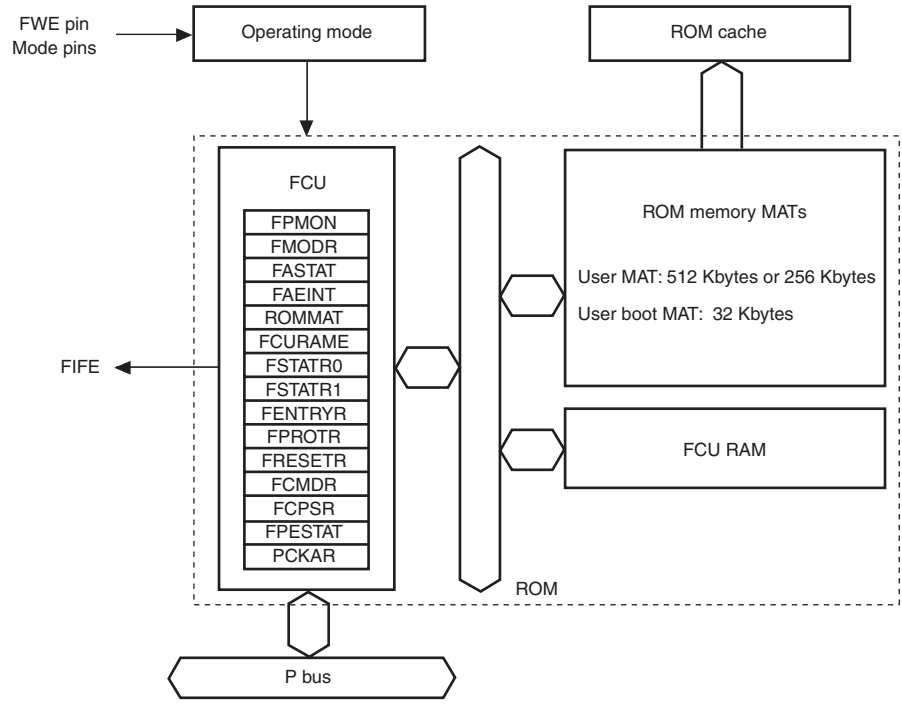
- High-speed reading through ROM cache

  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 23.2 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 23.3 shows the block configuration of the user MAT.

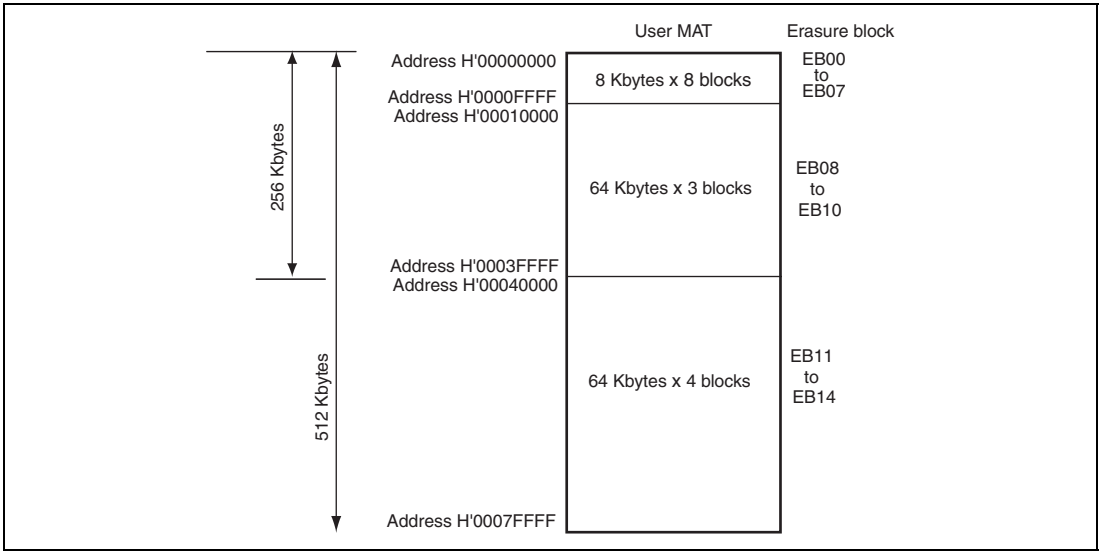


Figure 23.3 Block Configuration of User MAT

- Three types of on-board programming modes
  - Boot mode

    The user MAT and user boot MAT can be programmed using the SCIF. The bit rate for SCIF communications between the host and this LSI can be automatically adjusted.
  - 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 29, Electrical Characteristics.

23.2 Input/Output Pins

Table 23.1 shows the input/output pins used for the ROM. The combination of the FWE pin level and MD0 pin level determines the ROM programming mode (see section 23.4, Overview of ROM-Related Modes). In boot mode, the ROM can be programmed or erased by the host connected via the PB2/RxD3 and PB3/TxD3 pins (see section 23.5, Boot Mode).

Table 23.1 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	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 3	PB2/RxD3	Input	Receives data through SCIF channel 3 (communications with host)
Transmit data in SCI channel 3	PB3/TxD3	Output	Transmits data through SCIF channel 3 (communications with host)

23.3 Register Descriptions

Table 23.2 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 24.3, Register Descriptions. The ROM-related registers are initialized by a power-on reset.

Table 23.2 Register Configuration

Register Name	Symbol	R/W	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)*1	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)*2	H'0000 H'0001	H'FFFFA820	8, 16
FCU RAM enable register	FCURAME	R/(W)*2	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*4	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*4	H'FFFFA901	8
Flash P/E mode entry register	FENTRYR	R/(W)*3	H'0000*4	H'FFFFA902	8, 16
Flash protect register	FPROTR	R/(W)*3	H'0000*4	H'FFFFA904	8, 16
Flash reset register	FRESETR	R/(W)*2	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*4	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W	H'0000*4	H'FFFFA918	8, 16
Flash P/E status register	FPESTAT	R	H'0000*4	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*4	H'FFFFA938	8, 16

Notes: 1. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
2. 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.
3. 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.
4. These registers can be initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.

23.3.1 Flash Pin Monitor Register (FPMON)

FPMON monitors the FWE pin state. 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.

23.3.2 Flash Mode Register (FMODR)

FMODR specifies the FCU operation mode. 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 23.6.1, FCU Command List, and section 23.6.3 (13), Reading Lock Bit), whereas this bit must be set to make the blank check command available for use in the FLD (see section 24, 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.

23.3.3 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and FLD. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 23.9.3, 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 while the FENTRY0 bit in FENTRYR is 1 in ROM P/E normal mode. • An access command is issued to ROM program/erase addresses while the FENTRY0 bit in FENTRYR is 0. • A read access command is issued to ROM read addresses 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 23.9.3, 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 24, Data Flash (FLD).
2	EEPIFE	0	R/(W)*	FLD Instruction Fetch Error Refer to section 24, Data Flash (FLD).
1	EEPRPE	0	R/(W)*	FLD Read Protect Error Refer to section 24, Data Flash (FLD).
0	EEPWPE	0	R/(W)*	FLD Program/Erase Protect Error Refer to section 24, Data Flash (FLD).

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

23.3.4 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupts. 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 24, Data Flash (FLD).
2	EEPIFEIE	1	R/W	FLD Instruction Fetch Error Interrupt Enable Refer to section 24, Data Flash (FLD).
1	EEPRPEIE	1	R/W	FLD Read Protect Error Interrupt Enable Refer to section 24, 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 24, Data Flash (FLD).

23.3.5 ROM MAT Select Register (ROMMAT)

ROMMAT switches memory MATs in the ROM. 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.

23.3.6 FCU RAM Enable Register (FCURAME)

FCURAME enables or disables access to the FCU RAM area. 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.

23.3.7 Flash Status Register 0 (FSTATR0)

FSTATR0 indicates the FCU status. 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 24, 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 23.9.3, 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 23.9.3, 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 23.9.3, 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 23.6.4, 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 23.6.4, 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.

23.3.8 Flash Status Register 1 (FSTATR1)

FSTATR1 indicates the FCU status. 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.

23.3.9 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. 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 24, 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 ROM. 0: The ROM is in read mode 1: The ROM is in P/E mode Programming is enabled when the following conditions are all satisfied: • 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.

23.3.10 Flash Protect Register (FPROTR)

FPROTR enables or disables the protection function through the lock bits against programming and erasure. 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.

23.3.11 Flash Reset Register (FRESETR)

FRESETR is used for the initialization of FCU. 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

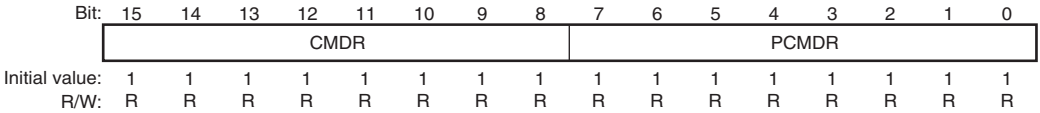
Note: * Write data is not retained.

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 29, 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.

23.3.12 FCU Command Register (FCMDR)

FCMDR stores the commands that the FCU has accepted. FCMDR is initialized by a power-on reset, or setting the FRESET bit of FRESETR to 1.



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 23.3 shows the states of FCMDR after acceptance of the various commands. For details on the blank check, see section 24.6, User Mode, User Program Mode, and User Boot Mode.

Table 23.3 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

23.3.13 FCU Processing Switch Register (FCPSR)

FCPSR selects a function to make the FCU suspend erasure. 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 23.6.4, Suspending Operation). 0: Suspension-priority mode 1: Erasure-priority mode

23.3.14 Flash P/E Status Register (FPESTAT)

FPESTAT indicates the result of programming/erasure of the ROM/FLD. 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

23.3.15 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.

23.3.16 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. 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 or 20 to 40 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.

23.4 Overview of ROM-Related Modes

Figure 23.4 shows the ROM-related mode transition in this LSI. For the relationship between the LSI operating modes and the MD0 and FWE pin settings, refer to section 3, MCU Operating Modes.

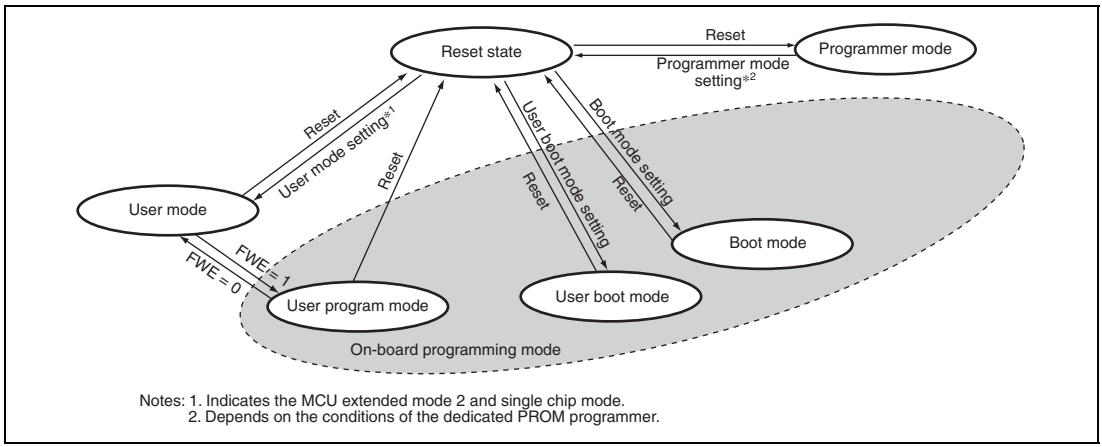


Figure 23.4 ROM-Related Mode Transition

- 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 23.4 compares programming- and erasure-related items for the boot mode, user program mode, user boot mode, and programmer mode.

Table 23.4 Comparison of Programming Modes

Item	Boot Mode	User Program Mode	User 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
Programming/ erasure control	Host	FCU	FCU	Programmer
Entire area erasure	Available (automatic)	Available	Available	Available (automatic)
Block erasure	Available*1	Available	Available	Not available
Programming data transfer	From host via SCIF	From any device via RAM	From any device via RAM	Via programmer
Reset-start MAT	Embedded program stored MAT	User MAT	User boot MAT*2	Embedded program stored MAT
Transition to MCU operating mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	—

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 Electronics 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, and programmer mode.
- In 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 SCIF. 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.

23.5 Boot Mode

23.5.1 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 SCIF 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 SCIF bit rate and performs communications between the host and this LSI by means of the control command method.

Figure 23.5 shows the system configuration in boot mode. The NMI and IRQ6 to 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.

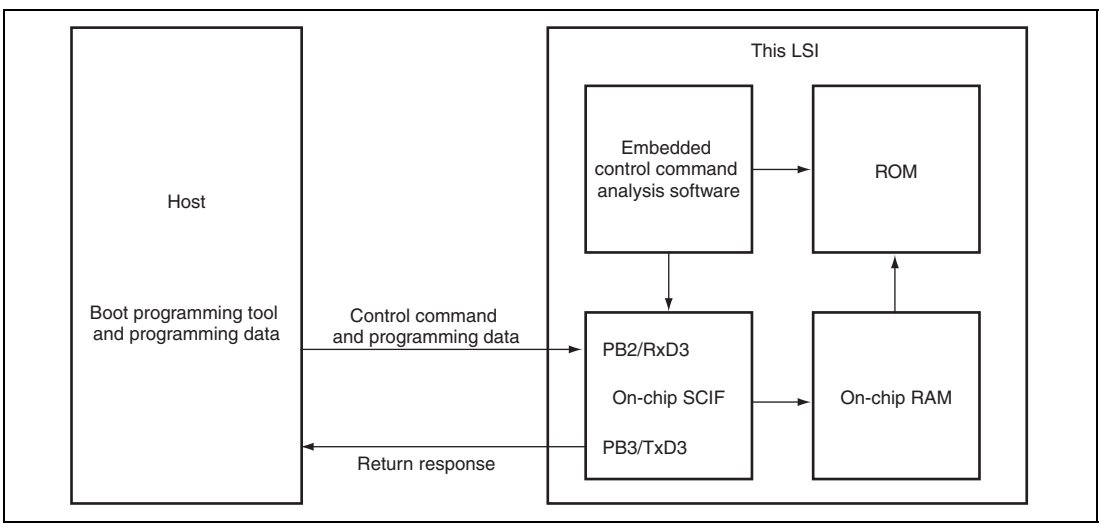


Figure 23.5 System Configuration in Boot Mode

23.5.2 State Transition in Boot Mode

Figure 23.6 shows the state transition in boot mode.

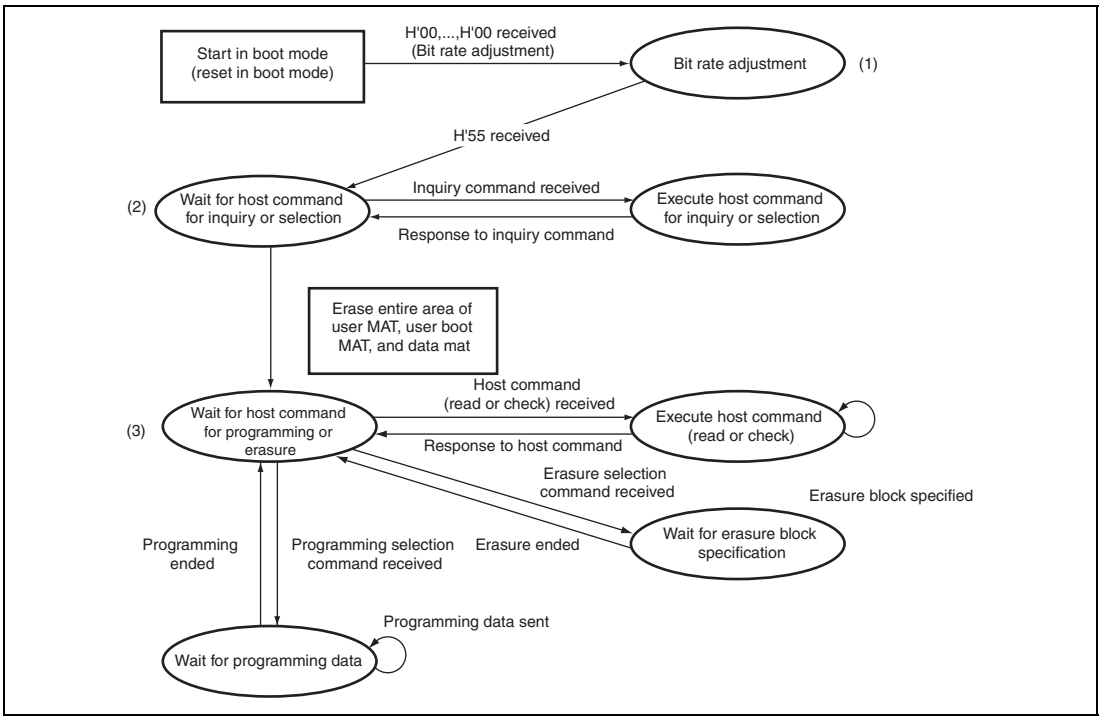


Figure 23.6 State Transition in Boot Mode

(1) Bit Rate Adjustment

After this LSI is started in boot mode, it automatically adjusts the bit rate for communications between the host and SCIF. 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 23.5.3, 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 23.5.4, 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 23.5.5, Programming/Erasing Host Command Wait State.

23.5.3 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 SCIF communications. During this measurement, set the SCIF 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 SCIF 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.

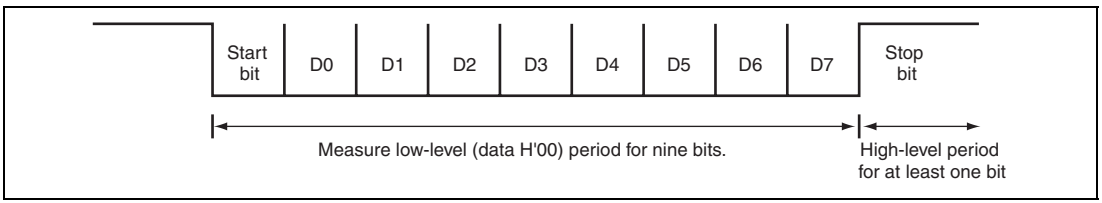


Figure 23.7 SCIF Transmit/Receive Format for Automatic Adjustment of Bit Rate

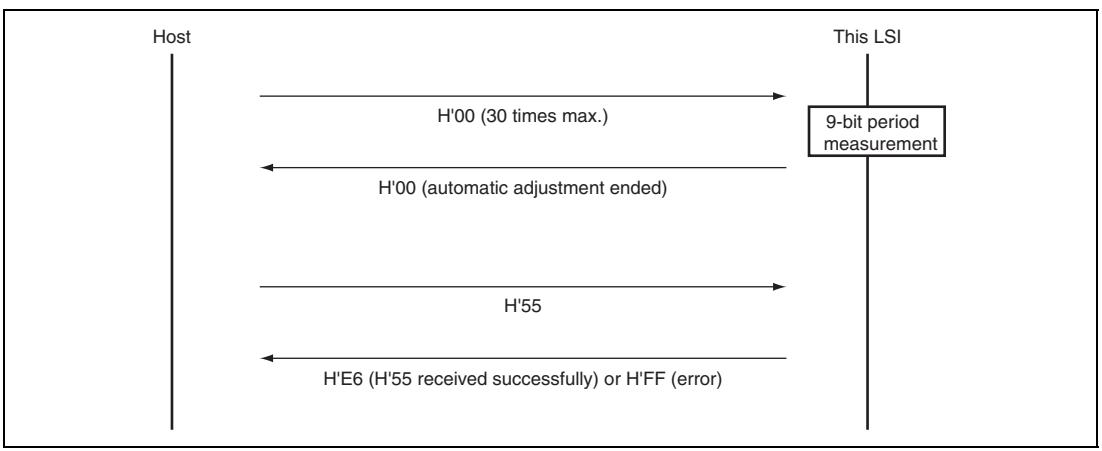


Figure 23.8 Communication Sequence between Host and This LSI

The bit rate may not be adjusted correctly depending on the bit rate of the host SCIF or the peripheral clock frequency of this LSI. Satisfy the SCIF communications condition as shown in table 23.5.

Table 23.5 Condition for Automatic Adjustment of Bit Rate

Host SCIF Bit Rate	Peripheral Clock Frequency of This LSI
9,600 bps	20 to 50 MHz
19,200 bps	20 to 50 MHz

23.5.4 Inquiry/Selection Host Command Wait State

Table 23.6 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 23.6 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 SCIF 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 23.9 shows an example of the procedure to use inquiry/selection host commands.

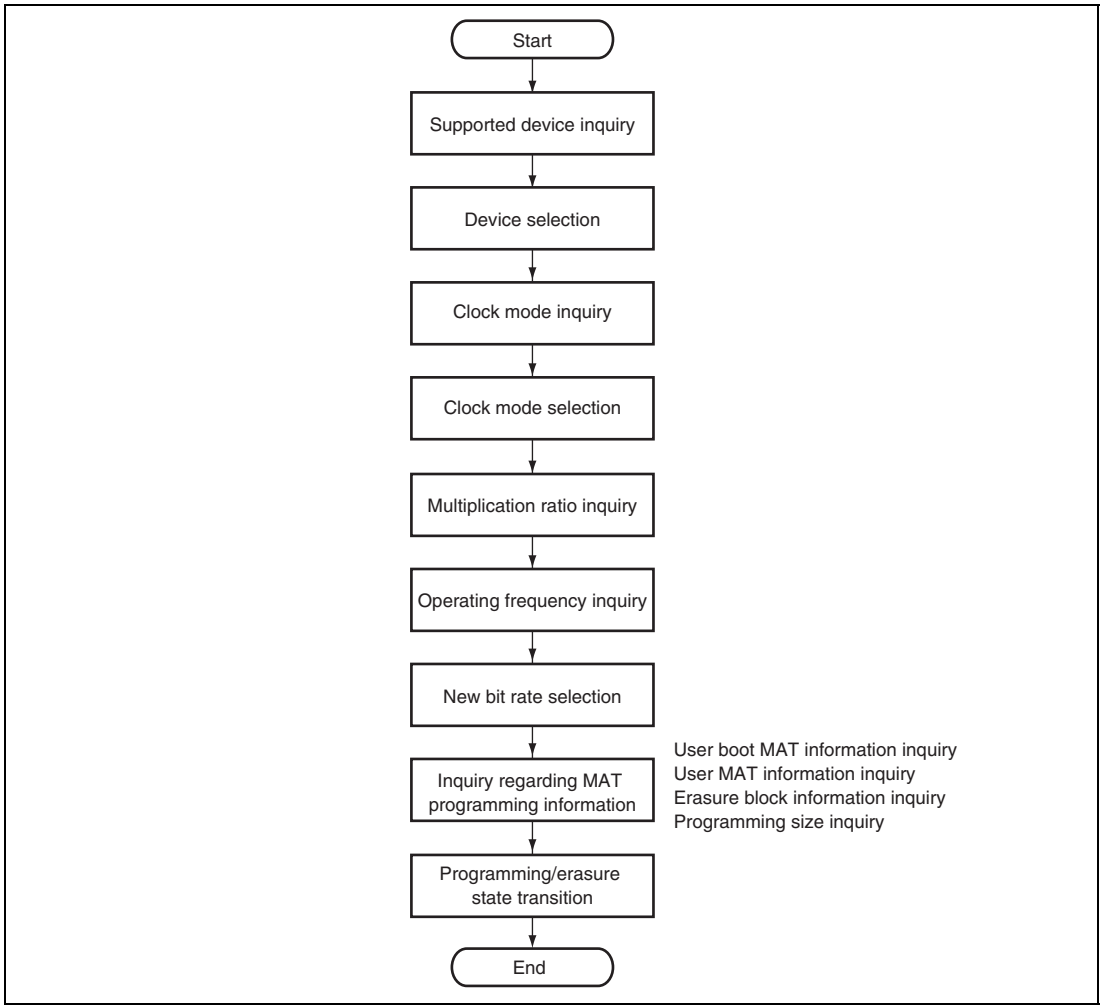


Figure 23.9 Example of Procedure to Use Inquiry/Selection Host Commands

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 SCIF can be set to the selected new bit rate. When the SCIF can be set to the new bit rate, this LSI returns a response (H'06) and sets the SCIF to the new bit rate. If the SCIF 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.

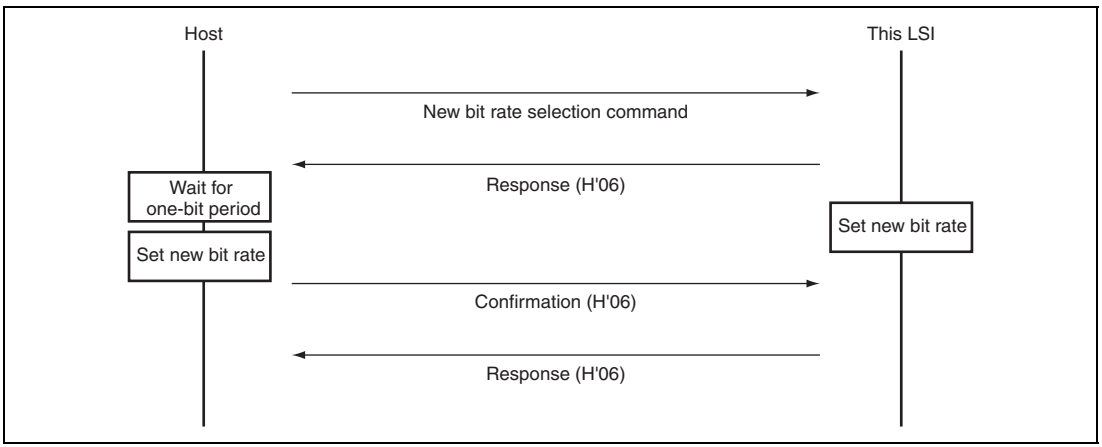


Figure 23.10 New Bit Rate Selection Sequence

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 SCIF 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 SCIF, 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 23.7)

Error (1 byte): Error status in this LSI (see table 23.8)

Table 23.7 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 23.8 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

23.5.5 Programming/Erasing Host Command Wait State

Table 23.9 shows the host commands available in programming/erasure host command wait state.

Table 23.9 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 23.5.4, 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 23.5.2, 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 23.5.2, 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.

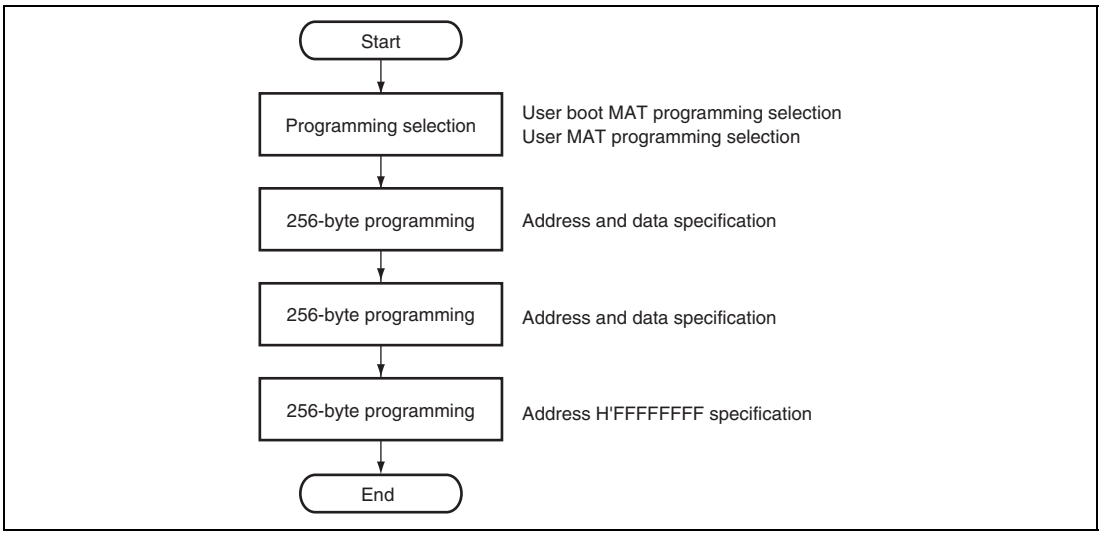


Figure 23.11 Procedure for ROM Programming in Boot Mode

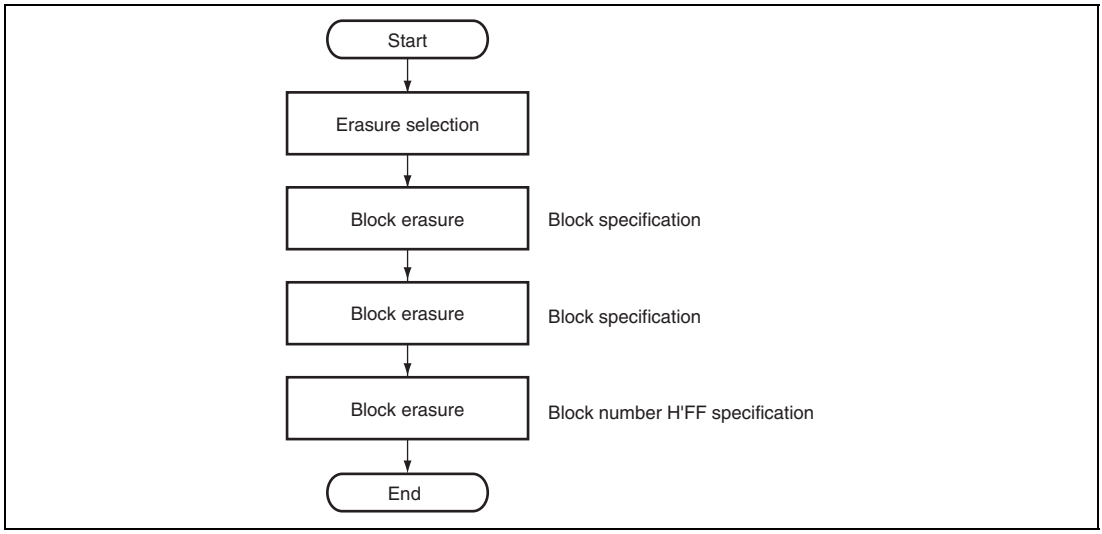


Figure 23.12 Procedure for ROM Erasure in Boot Mode

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 23.5.4, Inquiry/Selection Host Command Wait State.

23.6 User Program Mode

23.6.1 FCU Command List

To program or erase the user MAT in user program mode, issue FCU commands to the FCU. Table 23.10 is a list of FCU commands for ROM programming and erasure.

Table 23.10 FCU Command List (ROM-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 23.6.2, Conditions for FCU Command Acceptance)
Status read mode transition	Moves to the status read mode (see section 23.6.2, Conditions for FCU Command Acceptance)
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 23.6.2, 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	Notifies the sequencer of the frequency setting of the peripheral clock

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 24, Data Flash (FLD)).

To issue a command to the FCU, write to a ROM program/erase address through the P bus. Table 23.11 shows the FCU command format. Performing P-bus write access as shown in table 23.11 under specified conditions starts each command processing in the FCU. For the conditions for FCU command acceptance, refer to section 23.6.2, Conditions for FCU Command Acceptance. For details of each FCU command, refer to section 23.6.3, 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 23.6.4, Suspending Operation.

Table 23.11 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'8087FFFF

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)

23.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 23.13 is an FCU mode transition diagram.

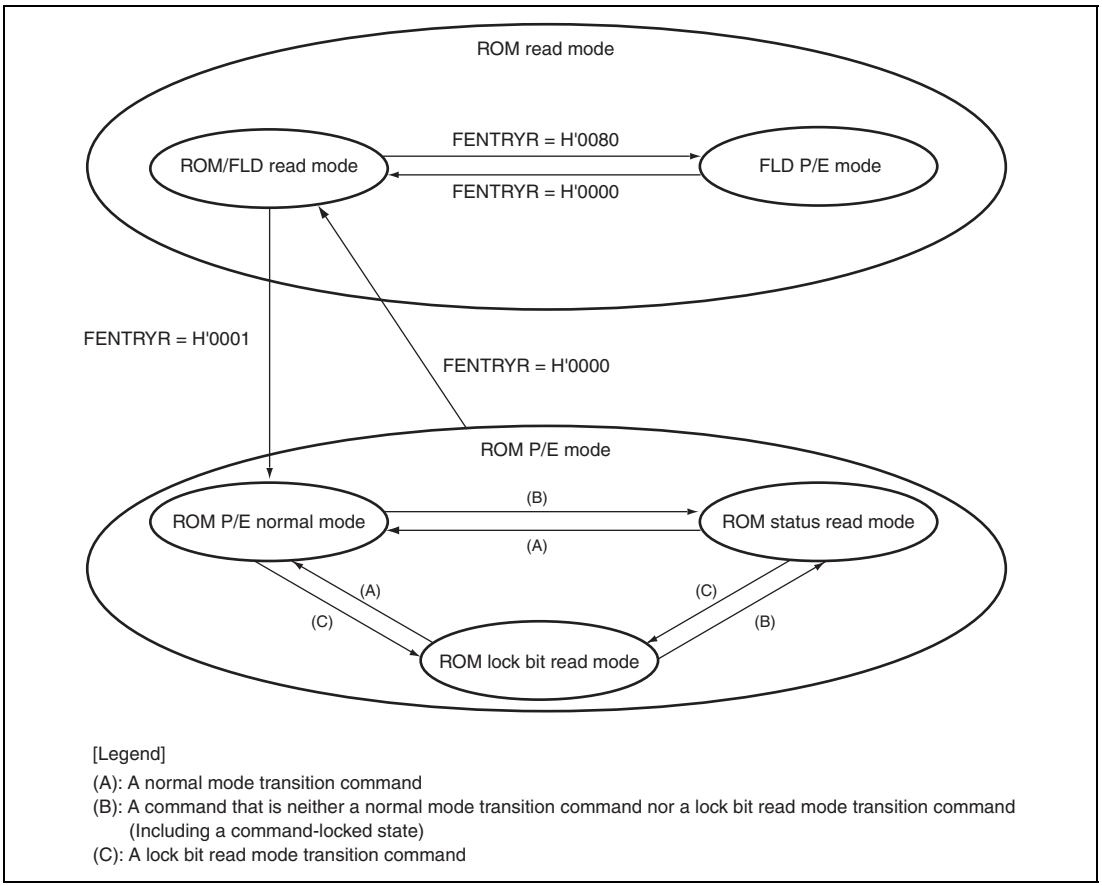


Figure 23.13 FCU Mode Transition Diagram (ROM-Related Modes)

(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 23.6.2, 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 23.12 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'8087FFFF 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 23.9.3, 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 23.12 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'8087FFFF 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 23.12 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'8087FFFF 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 23.12 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 23.9.3, 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 23.12, 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 23.12 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 23.12 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

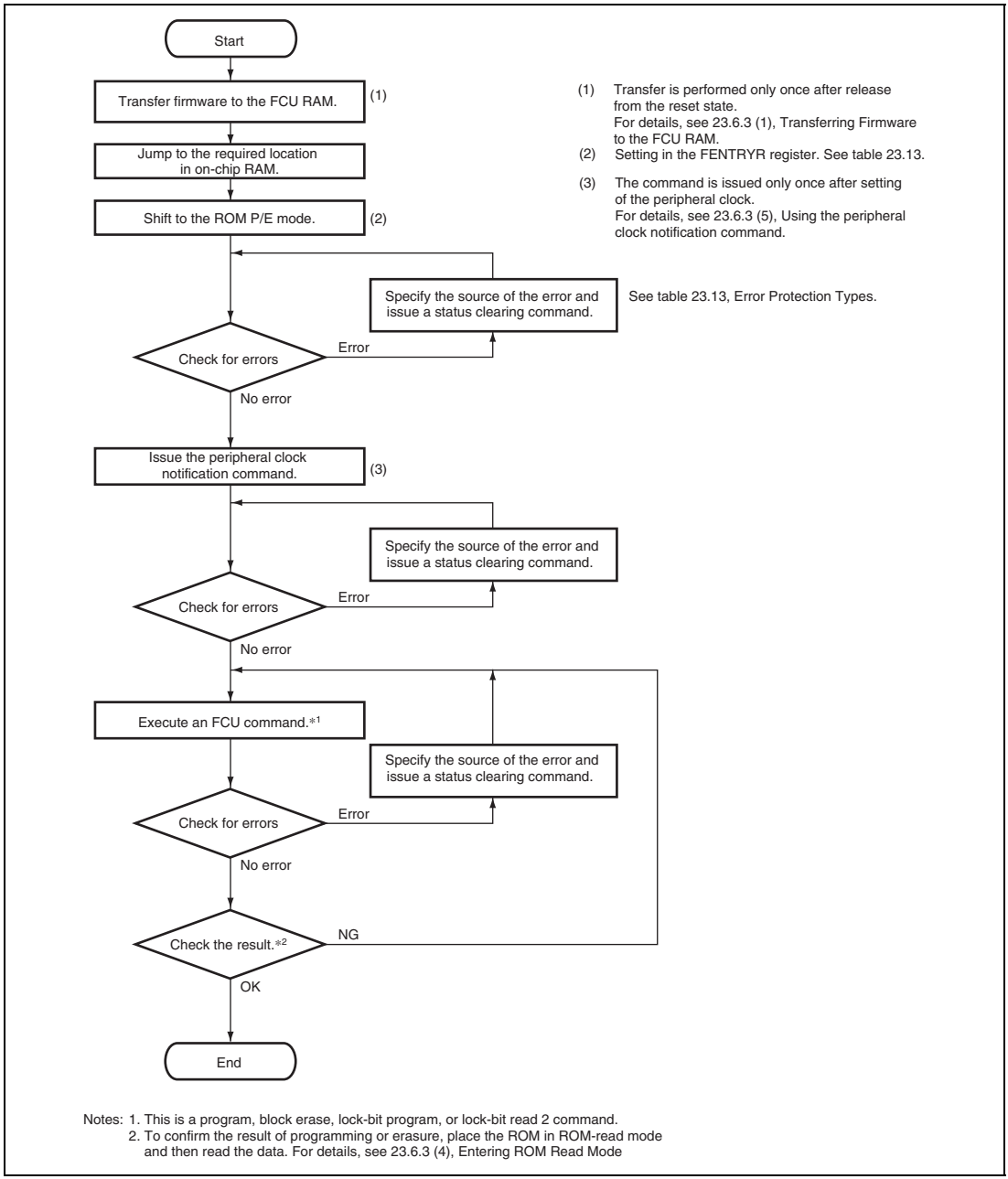
23.6.3 FCU Command Usage

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 23.6.2, 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 29, 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 23.14 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 23.6.3 (4), Entering ROM Read Mode

Figure 23.14 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 23.15 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 the firmware to the FCU RAM using the CPU.

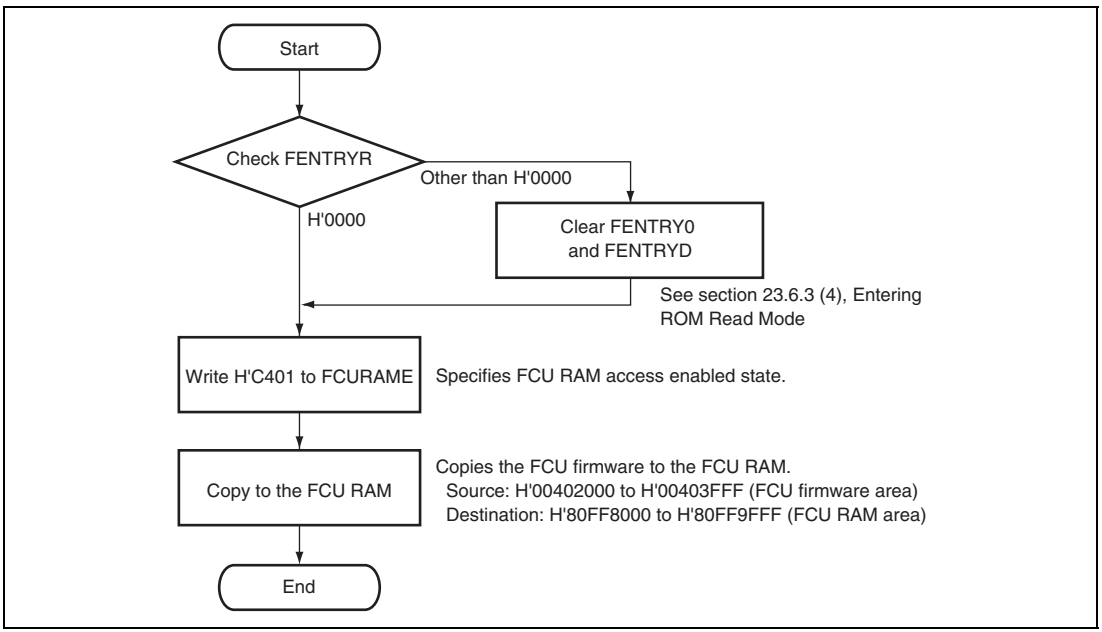


Figure 23.15 Procedure for Firmware Transfer to FCU RAM

(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 23.6.2, Conditions for FCU Command Acceptance). For the conditions for writing to the FENTRY0 bit, refer to section 23.3.10, 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.

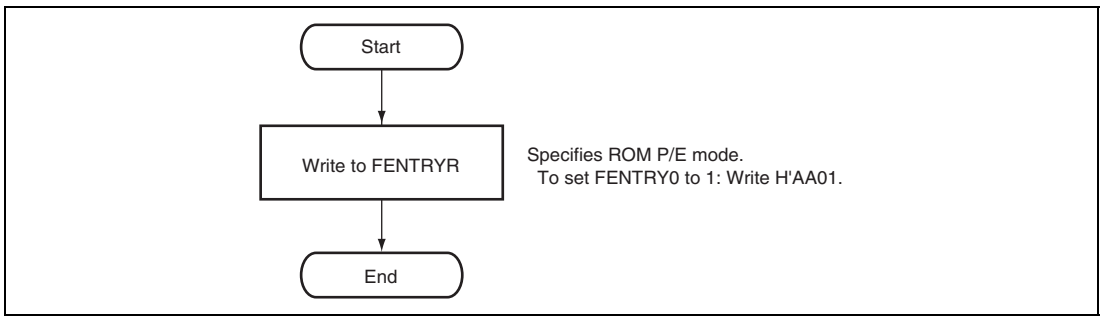
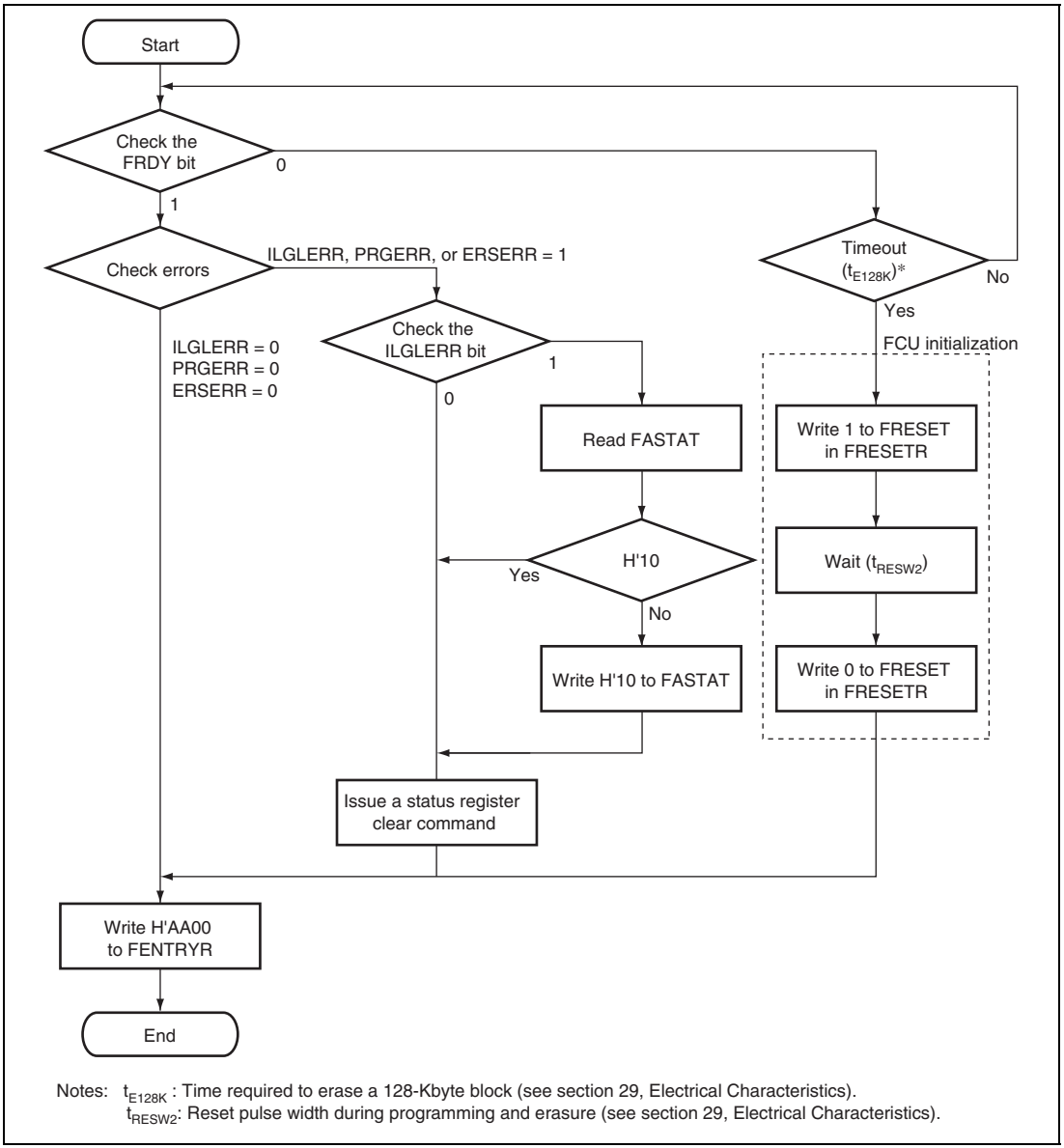


Figure 23.16 Procedure for Transition to ROM P/E Mode

(4) 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 23.6.2, 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 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

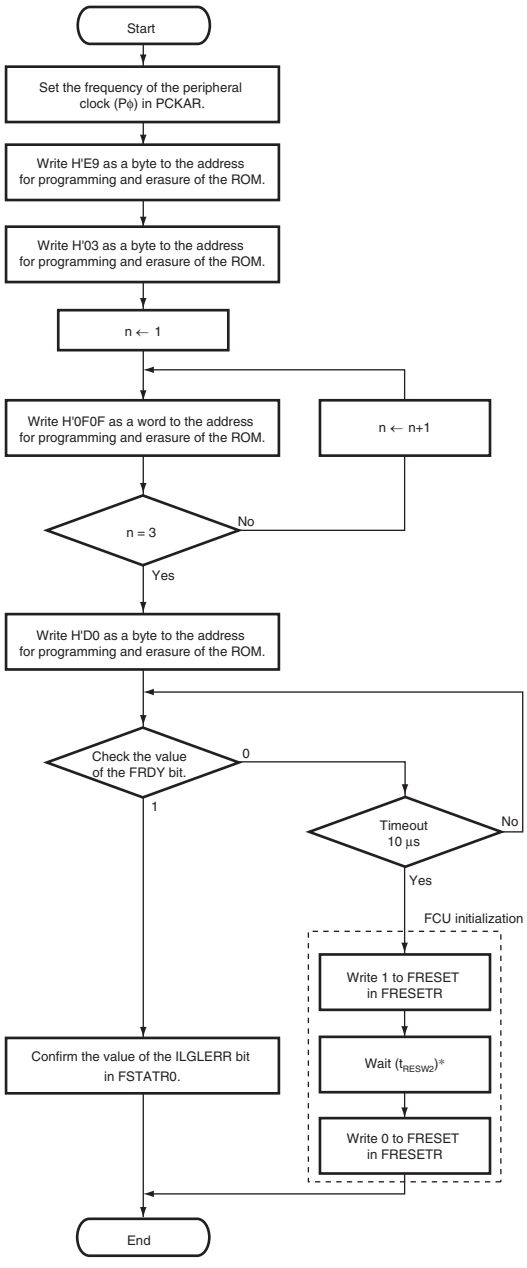
Figure 23.17 Procedure for Transition to ROM Read Mode

(5) 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 for the SH7239B and SH7237B, and from 20 to 40 MHz for the SH7239A and SH7237A. If the setting is not in this range, the FCU detects an error and enters the command-locked state (see section 23.9.3, 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.



Note: * t_{RESW2} denotes the width of a reset pulse during programming or erasure (see section 29, Electrical Characteristics).

Figure 23.18 Flow for Using the Peripheral Clock Notification Command

(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 23.6.3 (1), Transferring Firmware to the FCU RAM) and the other is to issue a normal mode transition command in ROM P/E mode (figure 23.19). 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.

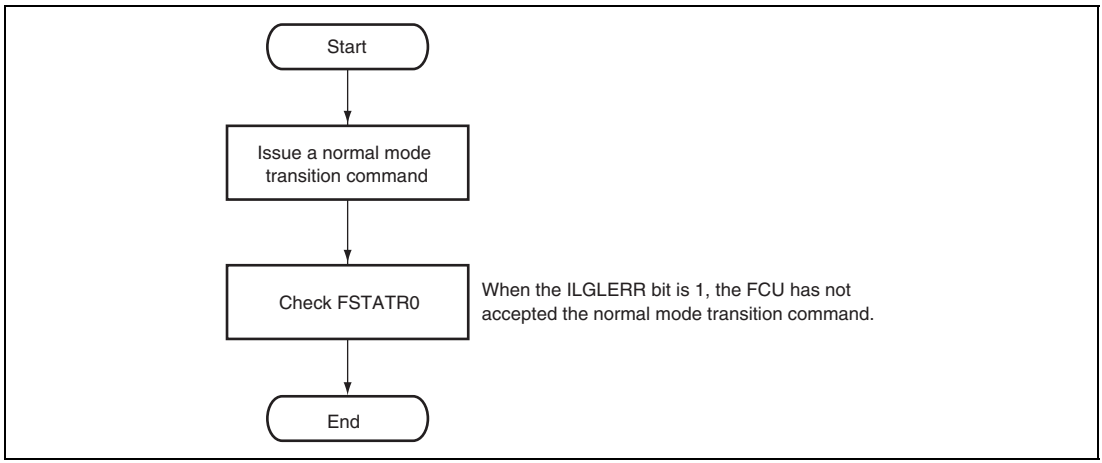
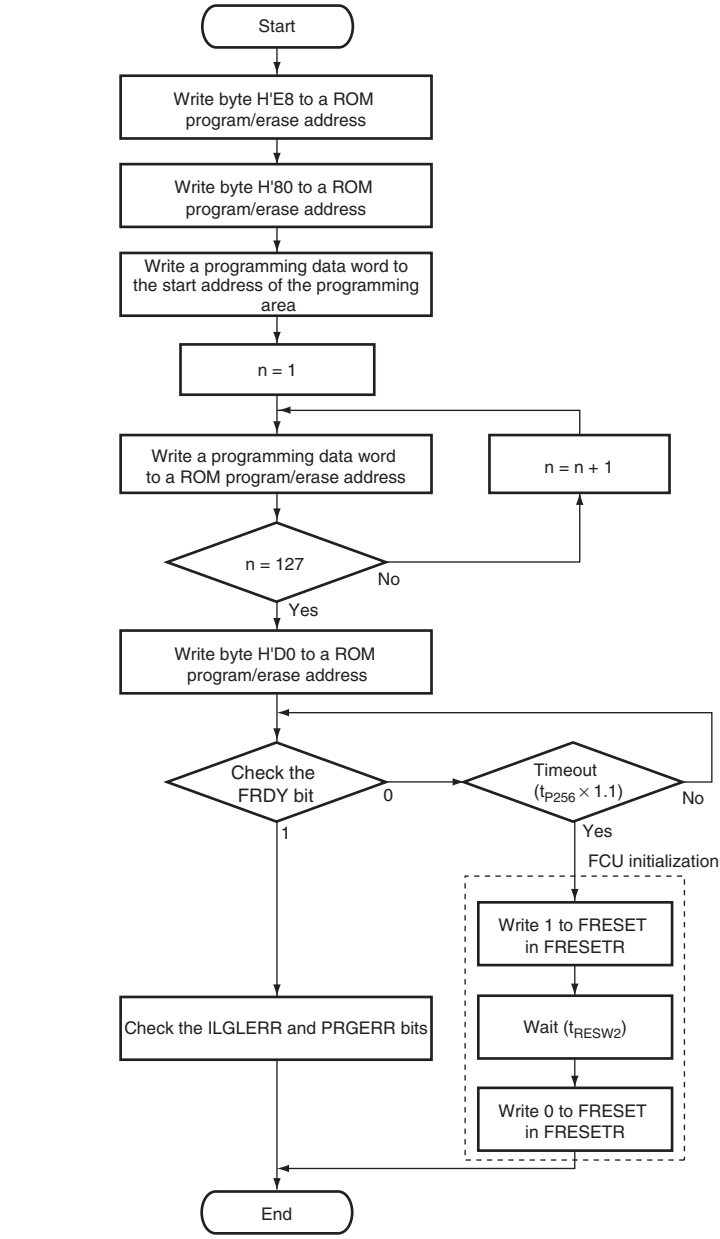


Figure 23.19 Procedure to Use ROM P/E Normal Mode Transition Command

(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 that, use word access to write 127 words to the ROM program/erase addresses.
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.



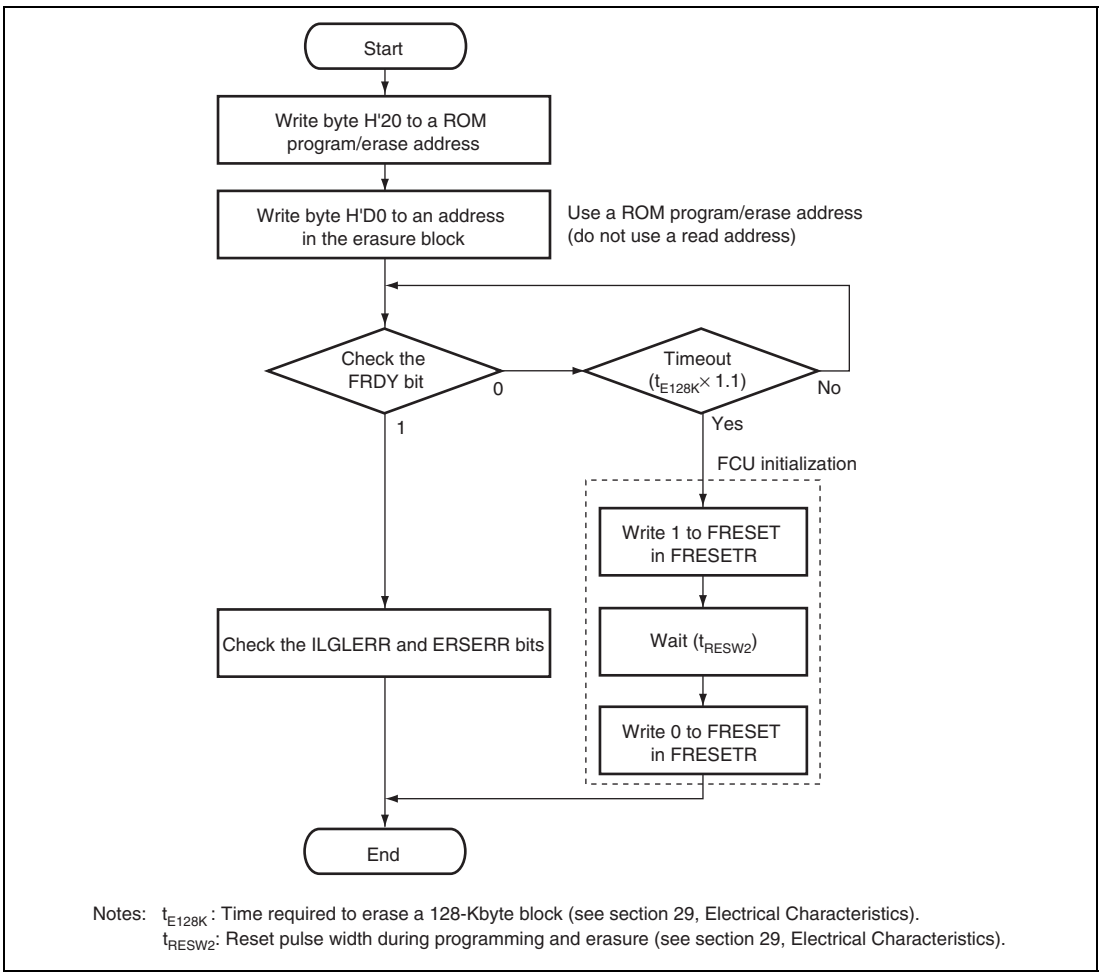
Notes: t_{P256}: Time required to write 256-byte data (see section 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 23.20 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.



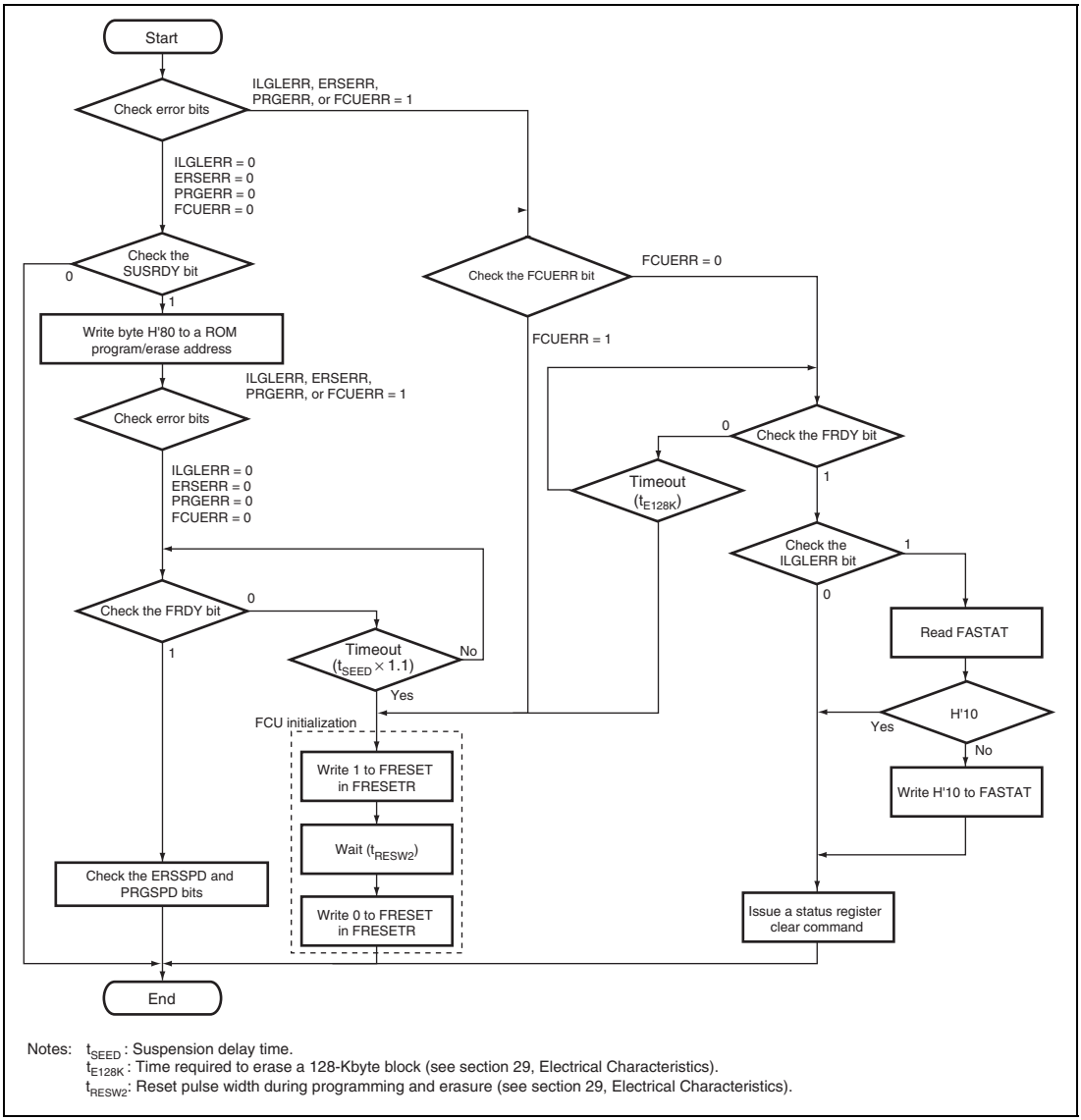
Notes: t_{E128K}: Time required to erase a 128-Kbyte block (see section 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 23.21 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 FSTATR0 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 23.9.3, Error Protection).



Notes: t_{SEED} : Suspension delay time.
t_{E128K} : Time required to erase a 128-Kbyte block (see section 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

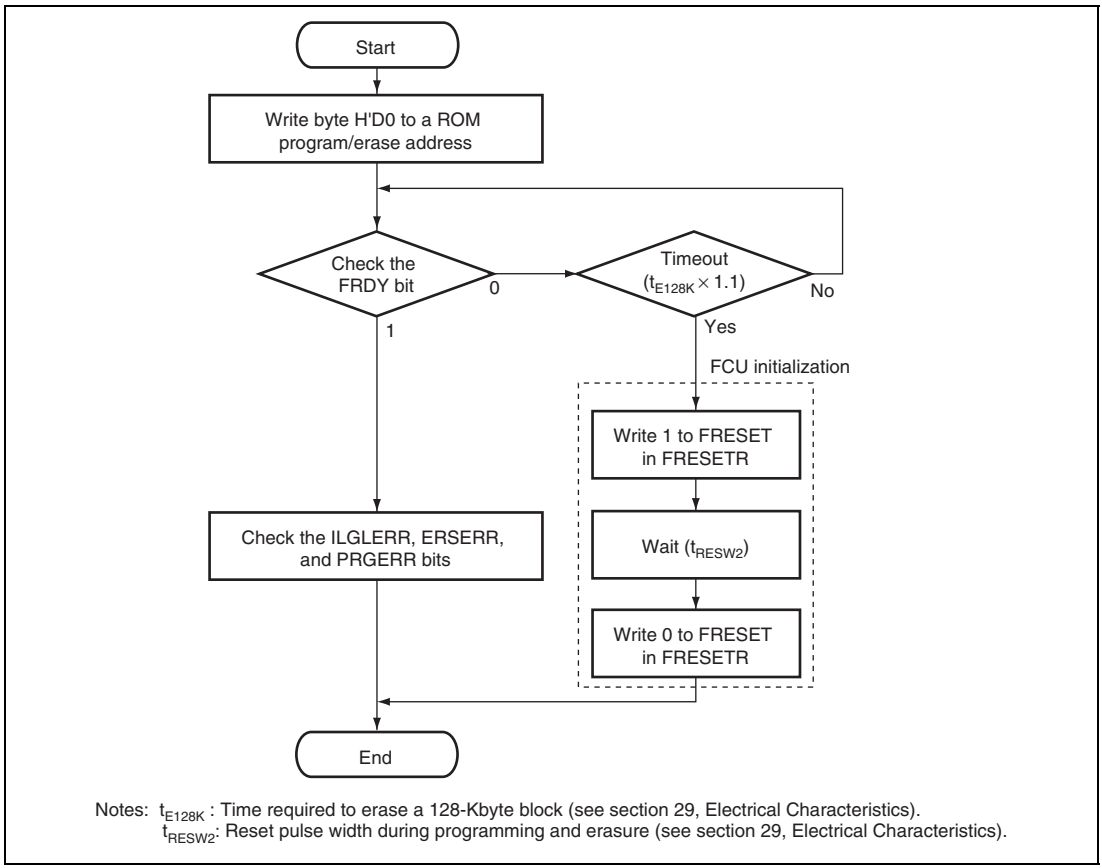
Figure 23.22 Procedure for Programming/Erasure Suspension

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 23.6.4, 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 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 23.23 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.

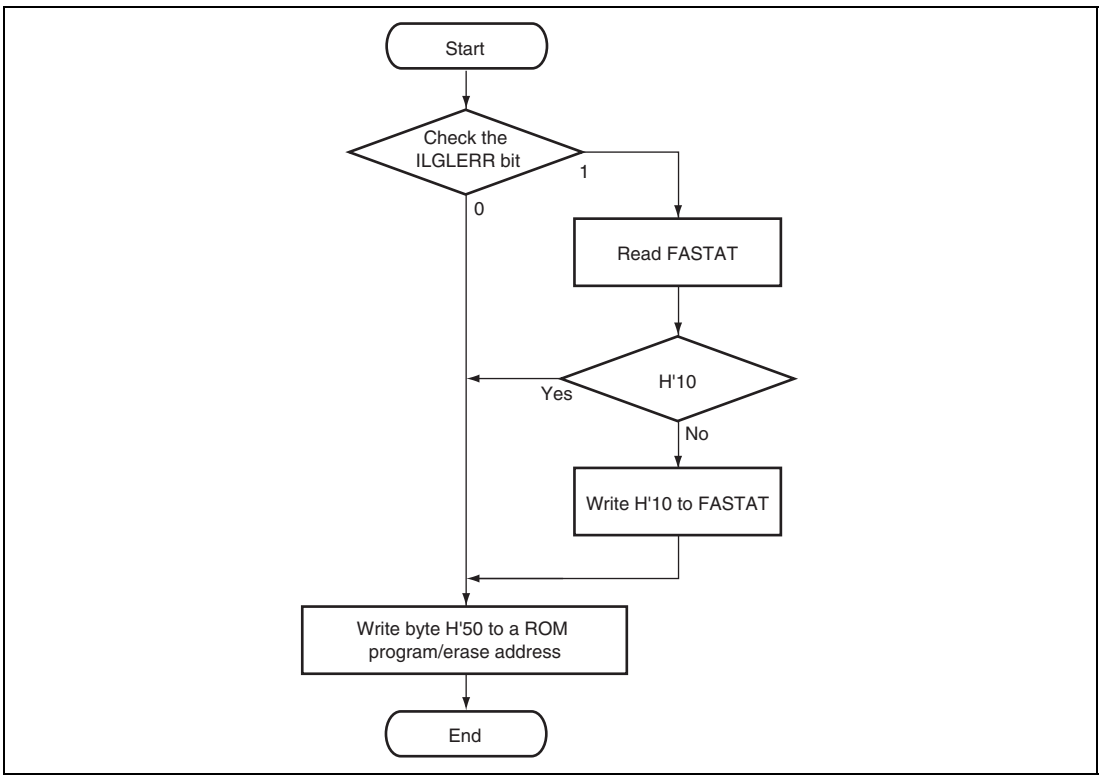


Figure 23.24 Procedure for Clearing Status Register 0

(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 23.25, 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.

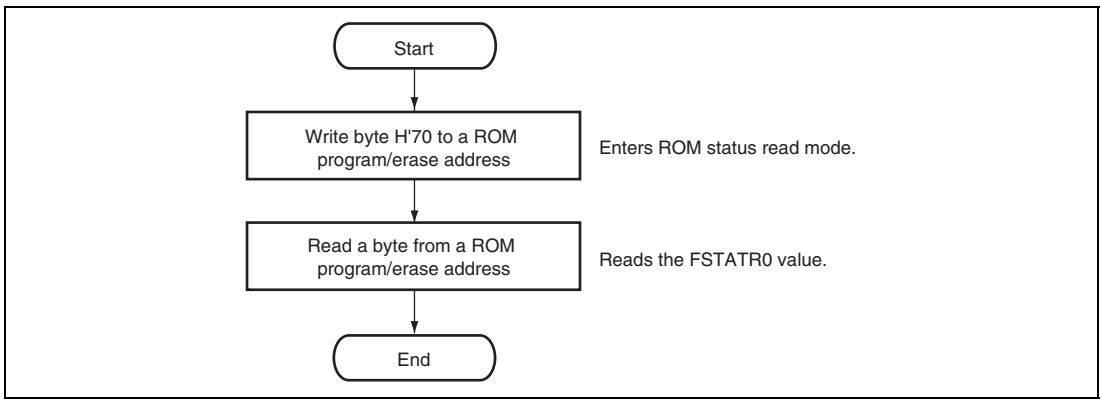


Figure 23.25 Procedure for Checking Status Register 0

(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.

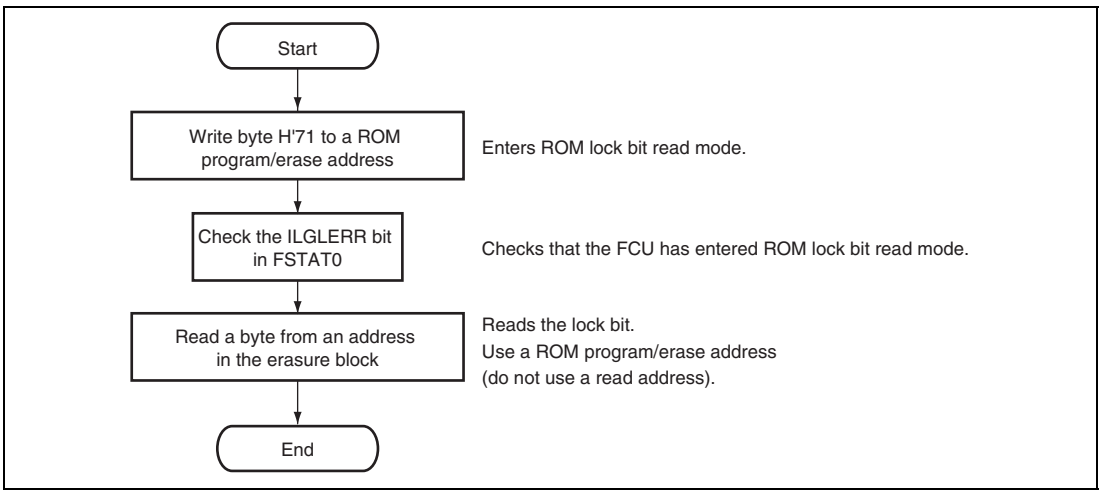
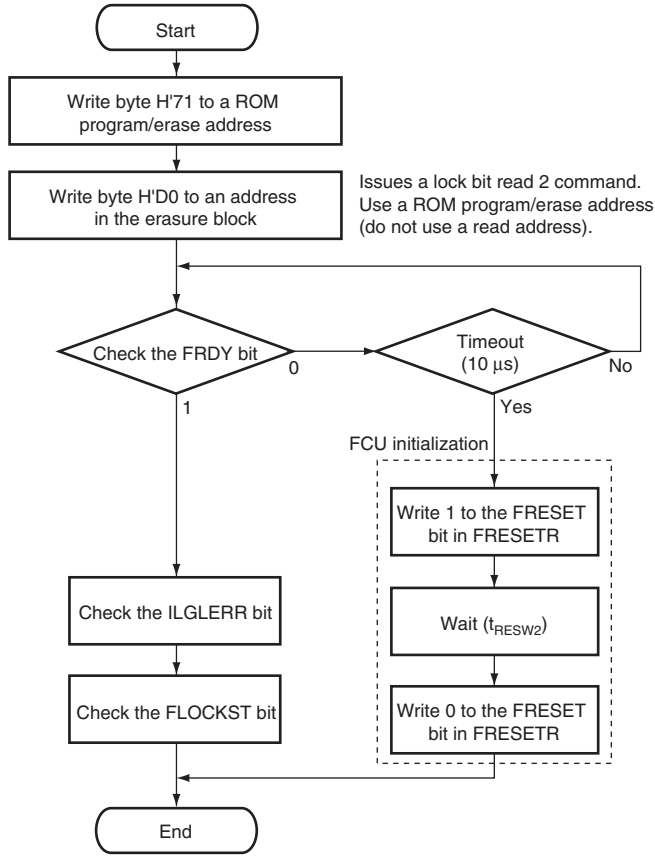


Figure 23.26 Procedure for Reading Lock Bit in Memory Area Read Mode

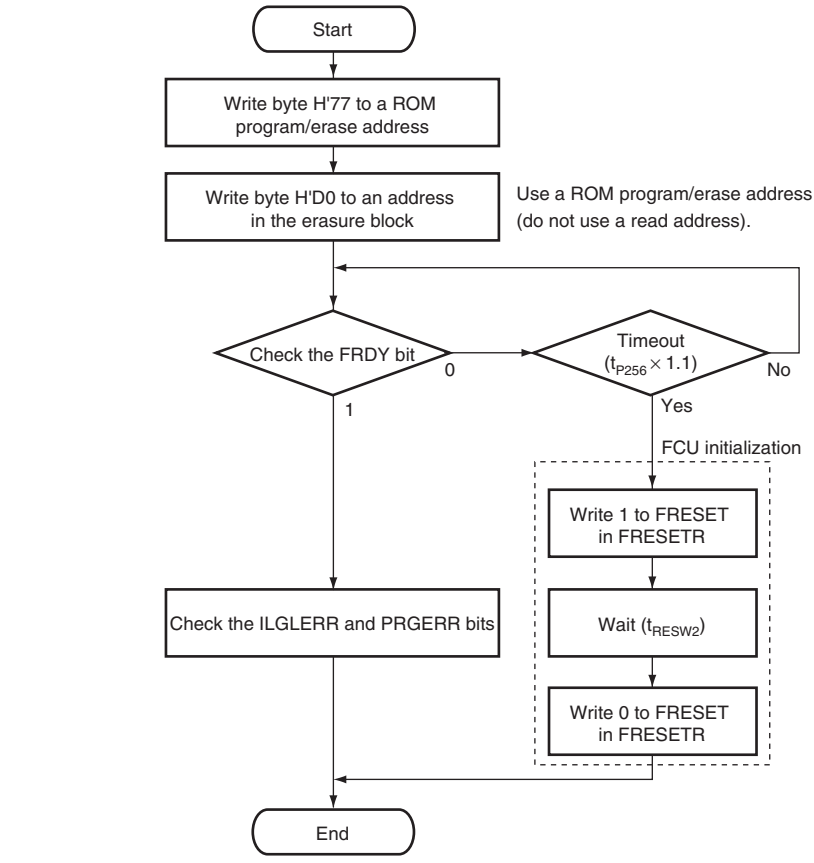


Note: t_{RESW2} : Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 23.27 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 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 23.28 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.

23.6.4 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 23.29 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.

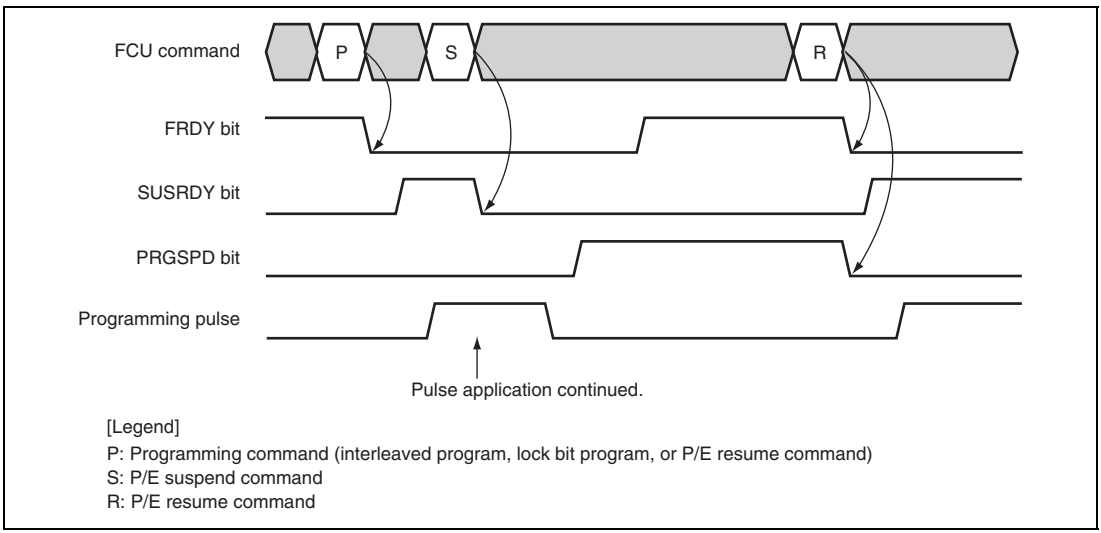


Figure 23.29 Suspending Programming Processing

Figure 23.30 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.

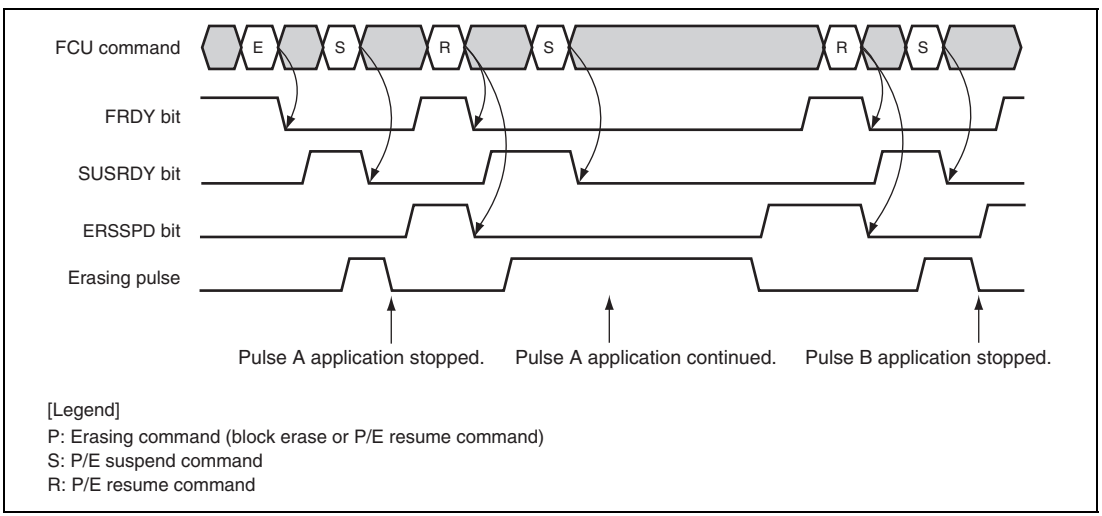


Figure 23.30 Suspending Erasure Processing (Suspension-Priority Mode)

Figure 23.31 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.

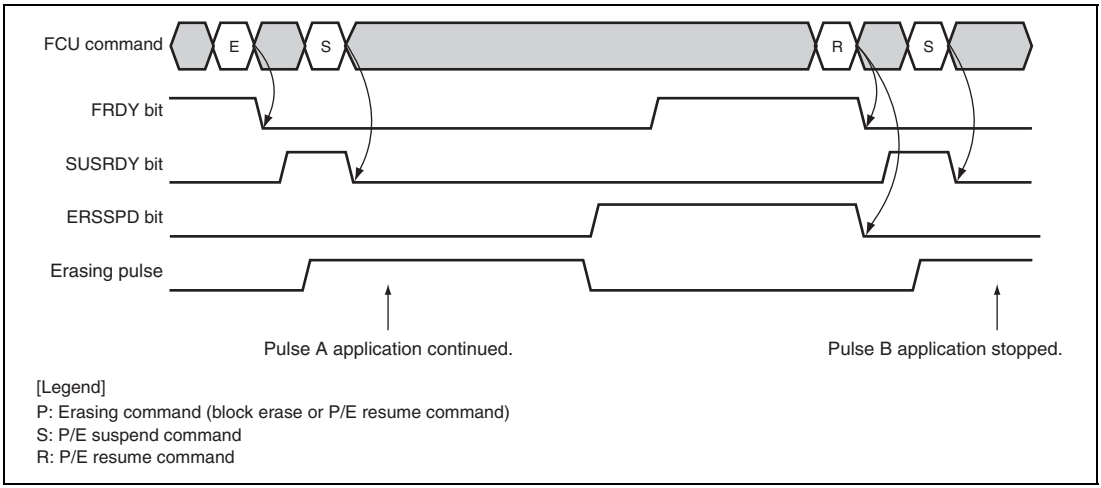


Figure 23.31 Suspending Erasure Processing (Erasure-Priority Mode)

23.7 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.

23.7.1 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 23.32 gives an overview of the boot sequence.

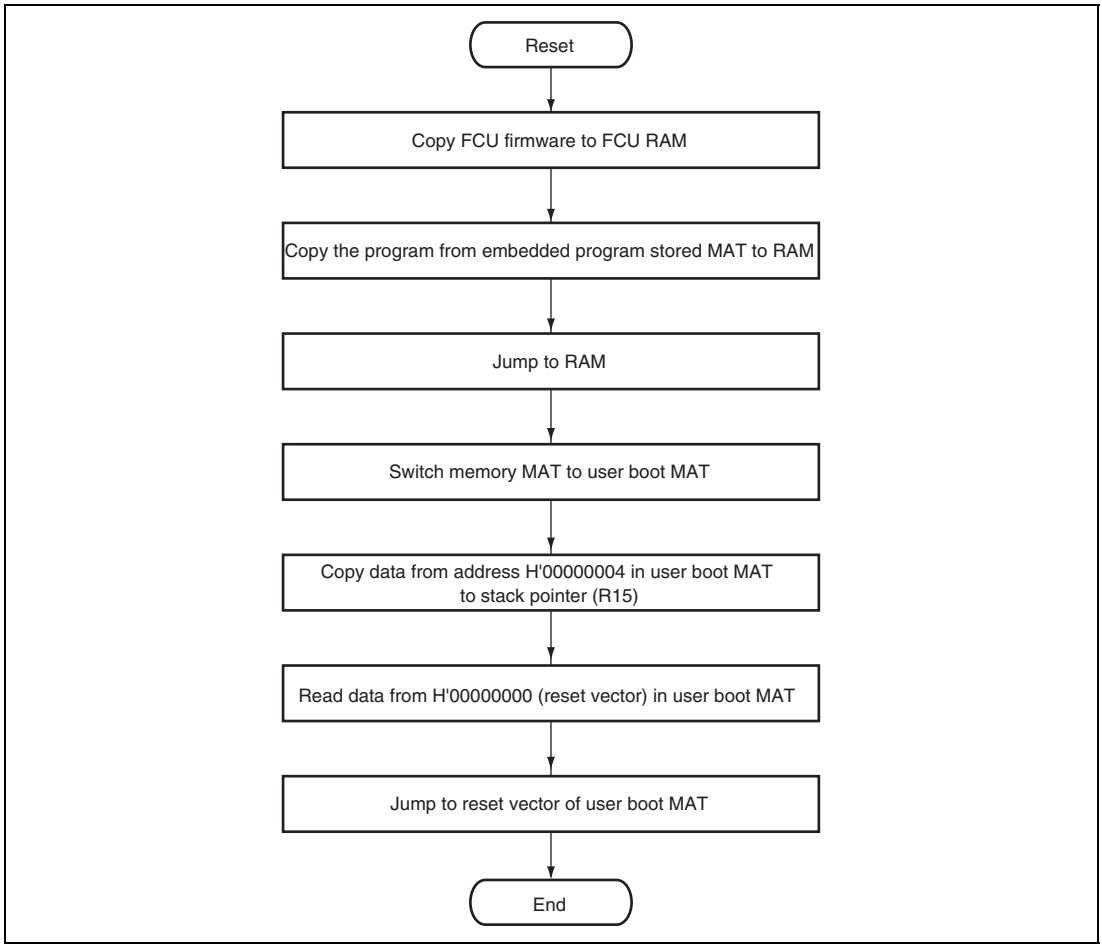


Figure 23.32 Overview of Boot Sequence in User Boot Mode

23.7.2 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 23.33 shows an example of the user MAT programming procedure.

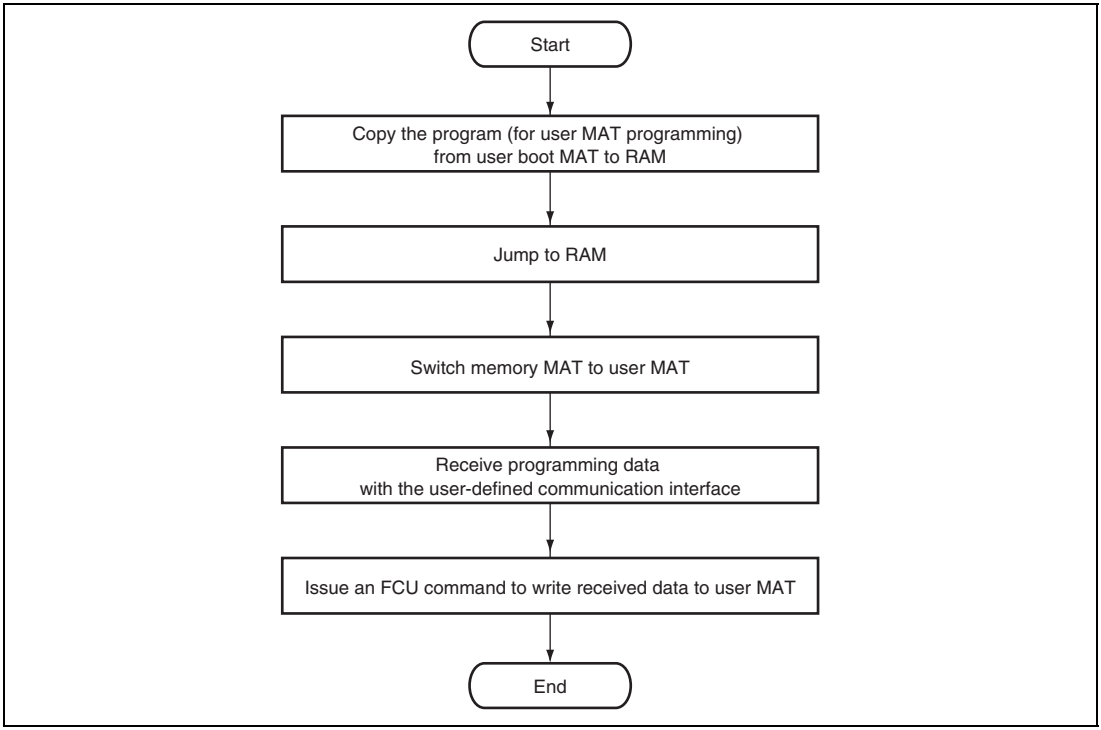


Figure 23.33 Example of User MAT Programming

23.8 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.

23.9 Protection

There are three types of ROM programming/erasure protection: hardware, software, and error protection.

23.9.1 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.

23.9.2 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'0007FFFF; program/erase addresses: H'80800000 to H'8087FFFF) 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 23.9.3, 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 23.9.3, Error Protection).

23.9.3 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 23.13 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 23.13 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 20 to 50 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'8087FFFF 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'8087FFFF while FENTRY0 = 0	1	0	0	0	1
	A read access command has been issued to addresses H'00000000 to H'0007FFFF 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

23.10 Usage Notes

23.10.1 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.

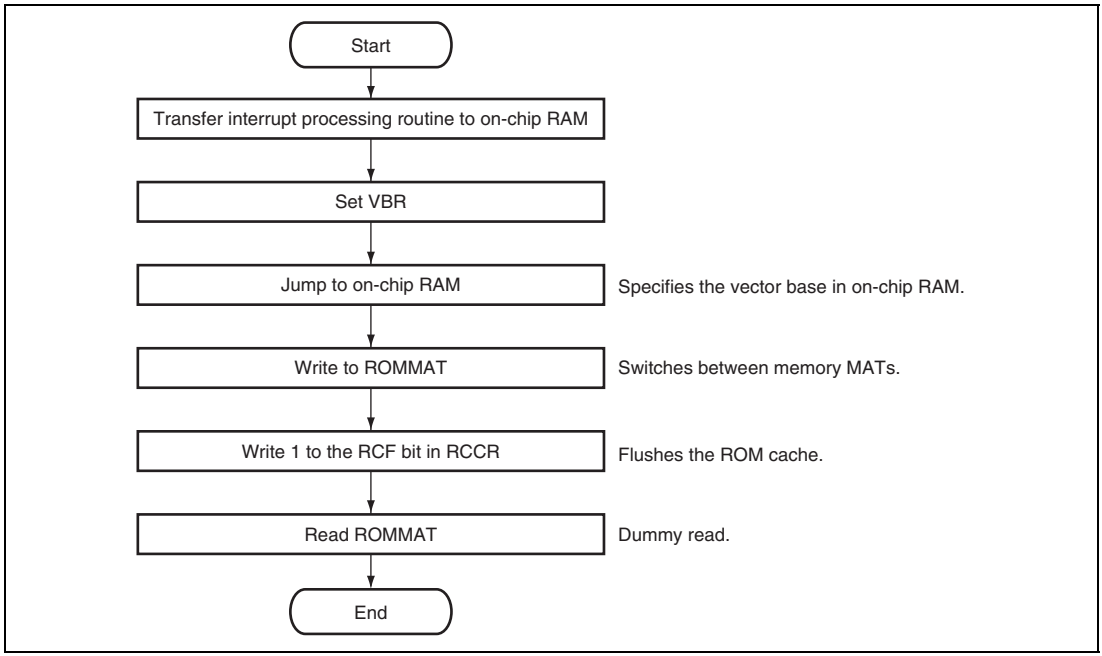


Figure 23.34 Example of MAT Switching Steps

23.10.2 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

23.10.3 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.

23.10.4 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.

23.10.5 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.

23.10.6 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 29, 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 29, 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.

23.10.7 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.

23.10.8 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.

23.10.9 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).

23.10.10 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

Section 24 Data Flash (FLD)

This LSI includes 32 Kbytes of flash memory (FLD) for storing data.

24.1 Features

- Flash-memory MATs
  Data MAT: 32 Kbytes (2 Kbytes × 16 blocks)

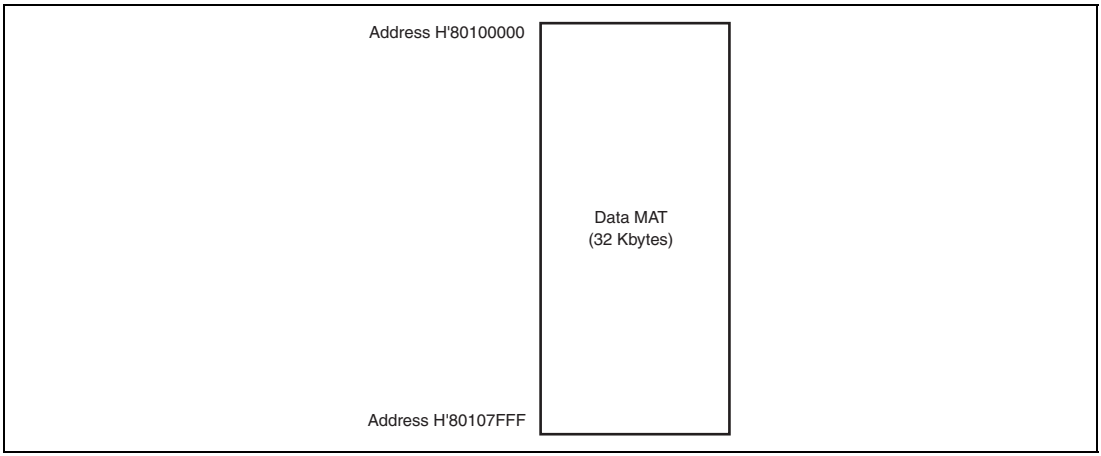


Figure 24.1 Memory MAT Configuration in FLD

- Reading through the peripheral bus (P bus)

  The data MAT can be read through the P bus.

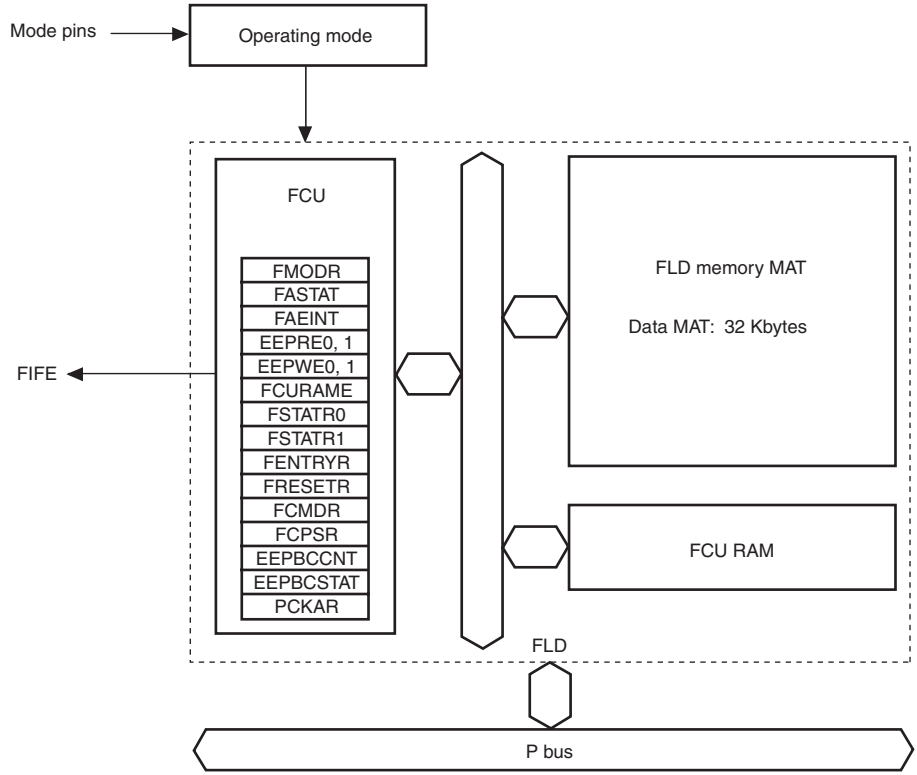
  Reading programs can be executed on the on-chip RAM or on-chip ROM.
- Programming and erasing methods

  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

  The CPU can execute programs located in areas other than the ROM while the FCU is programming or erasing the ROM.

  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, 1:	FLD read enable registers 0 and 1
EEPWE0, 1:	FLD program/erase enable registers 0 and 1
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 24.2 Block Diagram of FLD

- Programming/erasing unit

  The data MAT is programmed in 8-byte or 128-byte units and erased in block units (2 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 (2 Kbytes). The product information MAT is read-only memory and cannot be programmed or erased.

  Figure 24.3 shows the block configuration of the data MAT of this LSI. The data MAT is divided into sixteen 2-Kbyte blocks (DB00 to DB15).

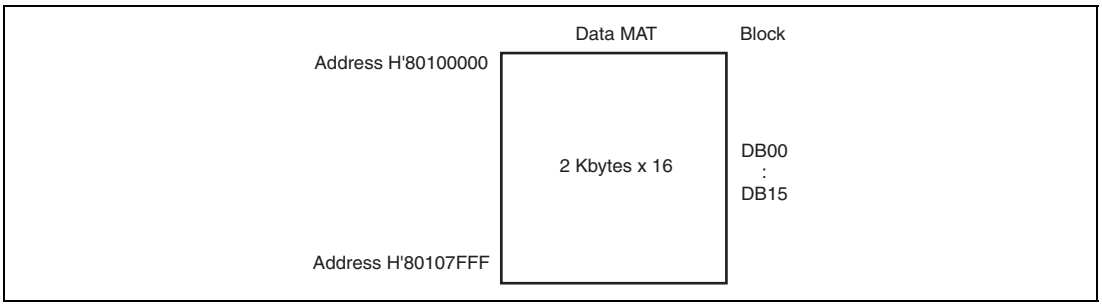


Figure 24.3 Block Configuration of Data MAT

- 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 a 2 Kbytes (1 erasure block) or 2 bytes of area can be checked by a single execution of the blank check command.

- Three types of on-board programming modes
  - — Boot mode

    The data MAT can be programmed using the SCIF. The bit rate for SCIF 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 2 and single-chip mode (modes 2 and 3) in which the on-chip ROM is enabled. However, note that mode 2 is not supported by the 5.0-V power supply voltage products (SH7239B and SH7237B).
  - — 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 in the FENTRYR register, EEPRE0 register, EEPRE1 register, EEPWE0 register, or EEPWE1 register. The FENTRYD bit in the FENTRYR register enables or disables data MAT programming or erasure by the FCU. EEPRE0 and EEPRE1 control protection of each data MAT block against reading, and EEPWE0 and EEPWE1 control 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 29, Electrical Characteristics.

24.2 Input/Output Pins

Table 24.1 shows the input/output pins used for the FLD. The combination of FWE and MD0 pin levels determines the FLD programming mode (see section 24.4, Overview of FLD-Related Modes). In boot mode, programming and erasing the FLD can be performed by the host via the PB2/RxD3 and PB3/TxD3 pins (refer to section 24.5, Boot Mode).

Table 24.1 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 SCIF channel 3	PB2/RxD3	Input	Receives data through SCIF channel 3 (communications with host)
Transmit data in SCIF channel 3	PB3/TxD3	Output	Transmits data through SCIF channel 3 (communications with host)

24.3 Register Descriptions

Table 24.2 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 23.3, Register Descriptions. The FLD-related registers are initialized by a power-on reset.

Table 24.2 Register Configuration

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Flash mode register	FMODR	R/W	H'00	H'FFFFA802	8
Flash access status register	FASTAT	R/(W)*1	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)*2	H'0000	H'FFFFA840	8, 16
FLD read enable register 1	EEPRE1	R/(W)*2	H'0000	H'FFFFA842	8, 16
FLD program/erase enable register 0	EEPWE0	R/(W)*2	H'0000	H'FFFFA850	8, 16
FLD program/erase enable register 1	EEPWE1	R/(W)*2	H'0000	H'FFFFA852	8, 16
FCU RAM enable register	FCURAME	R/(W)*2	H'0000	H'FFFFA854	8, 16
Flash status register 0	FSTATR0	R	H'80*4	H'FFFFA900	8, 16
Flash status register 1	FSTATR1	R	H'00*4	H'FFFFA901	8
Flash P/E mode entry register	FENTRYR	R/(W)*3	H'0000*4	H'FFFFA902	8, 16
Flash reset register	FRESETR	R/(W)*4	H'0000	H'FFFFA906	8, 16
FCU command register	FCMDR	R	H'FFFF*4	H'FFFFA90A	8, 16
FCU processing switch register	FCPSR	R/W	H'0000*4	H'FFFFA918	8, 16
FLD blank check control register	EEPBCCNT	R/W	H'0000*4	H'FFFFA91A	8, 16
FLD blank check status register	EEPBCSTAT	R	H'0000*4	H'FFFFA91E	8, 16
Peripheral clock notification register	PCKAR	R/W	H'0000*4	H'FFFFA938	8, 16

Notes: 1. This register consists of the bits where only 0 can be written to clear the flags and the read-only bits.
2. 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.
3. 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.
4. This register can be initialized by a power-on reset, or by setting the FRESET bit of FRESETR to 1.

24.3.1 Flash Mode Register (FMODR)

FMODR specifies an operating mode for the FCU. 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 24.6.1, FCU Command List, 24.6.3, FCU Command Usage), whereas this bit must be set to specify the read method for the lock bits in the ROM (see section 23, 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.

24.3.2 Flash Access Status Register (FASTAT)

FASTAT indicates the access error status for the ROM and FLD. If any bit in FASTAT is set to 1, the FCU enters command-locked state (see section 24.7.3, 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 23, 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 24.7.3, 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 and EEPRE1 settings 1: An attempt has been made to read data from the FLD against the EEPRE0 and EEPRE1 settings [Setting condition] • An attempt is made to read data from the FLD area that has been read-protected through the EEPRE0 and EEPRE1 settings. [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 and EEPWE1 settings. 0: No programming or erasing command has been issued to the FLD against the EEPWE0 and EEPWE1 settings 1: A programming or erasing command has been issued to the FLD against the EEPWE0 and EEPWE1 settings [Setting condition] • A programming or erasing command is issued to the FLD area that has been program/erase-protected through the EEPWE0 and EEPWE1 settings. [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.

24.3.3 Flash Access Error Interrupt Enable Register (FAEINT)

FAEINT enables or disables output of flash interface error (FIFE) interrupt requests. 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 23, 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

24.3.4 FLD Read Enable Register 0 (EEPRE0)

EEPRE0 enables or disables read access to blocks DB00 to DB07 (see figure 24.3) in the data MAT. 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								DBRE07	DBRE06	DBRE05	DBRE04	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/W	R/W	R/W	R/W	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 DBRE07 to DBRE00 bit modification. The data written to these bits are not stored.
7	DBRE07	0	R/W	DB07 to DB00 Block Read Enable Enables or disables read access to blocks DB07 to DB00 in the data MAT. The DBREi bit (i = 07 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
6	DBRE06	0	R/W	
5	DBRE05	0	R/W	
4	DBRE04	0	R/W	
3	DBRE03	0	R/W	
2	DBRE02	0	R/W	
1	DBRE01	0	R/W	
0	DBRE00	0	R/W	

Note: * Write data is not retained.

24.3.5 FLD Read Enable Register 1 (EEPRE1)

EEPRE1 enables or disables read access to blocks DB08 to DB15 (see figure 24.3) in the data MAT. EEPRE1 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								DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
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: * 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 DBRE15 to DBRE08 bit modification. The data written to these bits are not stored.
7	DBRE15	0	R/W	DB15 to DB08 Block Read Enable Enables or disables read access to blocks DB15 to DB08 in the data MAT. The DBREi bit (i = 15 to 08) controls read access to block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'D2 is written to the KEY bits. 0: Disables read access 1: Enables read access
6	DBRE14	0	R/W	
5	DBRE13	0	R/W	
4	DBRE12	0	R/W	
3	DBRE11	0	R/W	
2	DBRE10	0	R/W	
1	DBRE09	0	R/W	
0	DBRE08	0	R/W	

Note: * Write data is not retained.

24.3.6 FLD Program/Erase Enable Register 0 (EEPWE0)

EEPWE0 enables or disables programming and erasure of blocks DB00 to DB07 (see figure 24.3) in the data MAT. 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								DBWE07	DBWE06	DBWE05	DBWE04	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/W	R/W	R/W	R/W	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 DBWE07 to DBWE00 bit modification. The data written to these bits are not stored.
7	DBWE07	0	R/W	DB07 to DB00 Block Program/Erase Enable Enables or disables programming and erasure of blocks DB07 to DB00 in the data MAT. The DBWEi bit (i = 07 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
6	DBWE06	0	R/W	
5	DBWE05	0	R/W	
4	DBWE04	0	R/W	
3	DBWE03	0	R/W	
2	DBWE02	0	R/W	
1	DBWE01	0	R/W	
0	DBWE00	0	R/W	

Note: * Write data is not retained.

24.3.7 FLD Program/Erase Enable Register 1 (EEPWE1)

EEPWE1 enables or disables programming and erasure of blocks DB15 to DB08 (see figure 24.3) in the data MAT. EEPWE1 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								DBWE15	DBWE14	DBWE13	DBWE12	DBWE11	DBWE10	DBWE09	DBWE08
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: * 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 DBWE15 to DBWE08 bit modification. The data written to these bits are not stored.
7	DBWE15	0	R/W	DB15 to DB08 Block Program/Erase Enable Enables or disables programming and erasure of blocks DB15 to DB08 in the data MAT. The DBWEi bit (i = 15 to 08) controls programming and erasure of block DBi. Writing to these bits is enabled only when this register is accessed in word size and H'E1 is written to the KEY bits. 0: Disables programming and erasure 1: Enables programming and erasure
6	DBWE14	0	R/W	
5	DBWE13	0	R/W	
4	DBWE12	0	R/W	
3	DBWE11	0	R/W	
2	DBWE10	0	R/W	
1	DBWE09	0	R/W	
0	DBWE08	0	R/W	

Note: * Write data is not retained.

24.3.8 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. 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 read 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 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 23, Flash Memory (ROM).

Note: * Write data is not retained.

24.3.9 FLD Blank Check Register (EEPBCCNT)

EEPBCCNT specifies the addresses and sizes of the target areas to be checked by the blank check command. 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.
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 2 Kbytes as the size of a blank check target area.

24.3.10 FLD Blank Check Status Register (EEPBCSTAT)

EEPBCSTAT stores check results by executing the blank check command. 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.

24.4 Overview of FLD-Related Modes

Figure 24.4 shows the FLD-related mode transition in this LSI. For the relationship between the LSI operating modes and MD0 and FWE pin settings, refer to section 3, MCU Operating Modes.

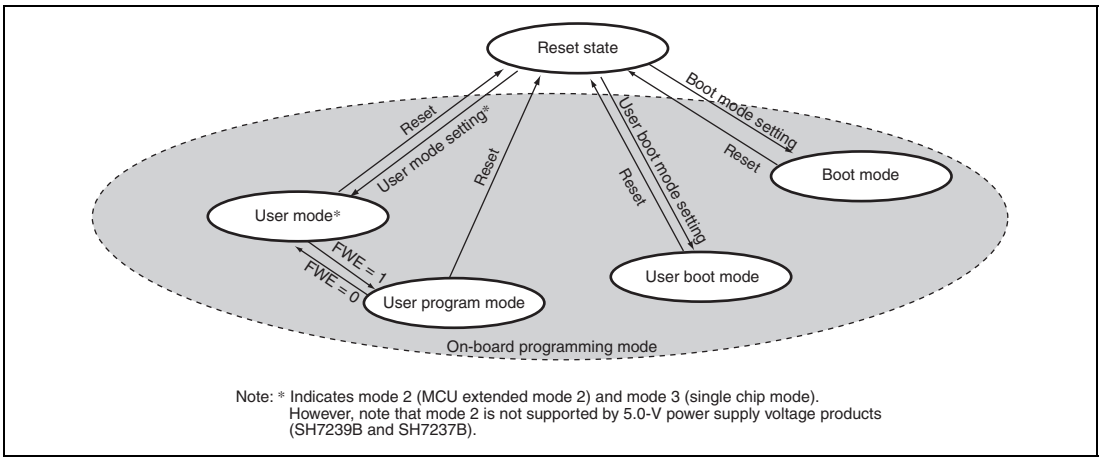


Figure 24.4 FLD-Related Mode Transition

- The data MAT can be read, programmed, and erased on the board in user mode, user program mode, user boot mode, and 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 24.3 compares programming- and erasure-related items for the boot mode, user mode, user program mode, and user boot mode.

Table 24.3 Comparison of Programming Modes

Item	Boot Mode	User Mode	User Program Mode	User Boot Mode
Programming/ erasure environment	On-board programming			
Programming/ erasure enabled MAT	Data MAT	Data MAT	Data MAT	Data MAT
Programming/ erasure control	Host	FCU	FCU	FCU
Entire area erasure	Available (automatic)	Available	Available	Available
Block erasure	Available*1	Available	Available	Available
Programming data transfer	From host via SCI	From any device via RAM	From any device via RAM	From any device via RAM
Reset-start MAT	Embedded program stored MAT	User MAT	User MAT	User boot MAT*2

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 Electronics Corp. is executed, execution starts from the location indicated by the reset vector of the user boot MAT.

- In 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 SCIF. 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.

24.5 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 23, Flash Memory (ROM). This section describes only the commands dedicated for the FLD.

24.5.1 Inquiry/Selection Host Commands

Table 24.4 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 23.11 in section 23.5.4, Inquiry/Selection Host Command Wait State.

Table 24.4 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 23.5.4, Inquiry/Selection Host Command Wait State).

24.5.2 Programming/Erasing Host Commands

Table 24.5 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 23.5.5, Programming/Erasing Host Command Wait State. For the erasure block information inquiry command, refer to section 23.5.4, Inquiry/Selection Host Command Wait State.

Table 24.5 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

24.6 User Mode, User Program Mode, and User Boot Mode

24.6.1 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 24.6 is a list of FCU commands for FLD programming and erasure.

Table 24.6 FCU Command List (FLD-Related Commands)

Command	Function
Normal mode transition	Moves to the normal mode (see section 24.6.2, Conditions for FCU Command Acceptance).
Status read mode transition	Moves to the status read mode (see section 24.6.2, Conditions for FCU Command Acceptance).
Lock bit read mode transition (lock bit read 1)	Moves to the lock bit read mode (see section 24.6.2, 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	Notifies the sequencer of the frequency setting of the peripheral clock.

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 24.7 shows the FCU command formats for the program command and blank check command. For the other command formats, refer to section 23.6.1, FCU Command List. When a P-bus access, as shown in table 24.7, 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 24.6.2, Conditions for FCU Command Acceptance. For details of command usage, refer to section 24.6.3, FCU Command Usage.

When the FRDMD bit in the FMODR register 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 in the FMODR register 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 23.6.4, Suspending Operation.

Table 24.7 FCU Command Formats (for FLD only)

	Number of Bus	First Cycle		Second Cycle		Third Cycle		Fourth Cycle to Cycle N + 2		Cycle N + 3	
Command	Cycles	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'80100000 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)

24.6.2 Conditions for FCU Command Acceptance

The FCU determines whether to accept a command depending on the FCU mode or status. Figure 24.5 is an FCU mode transition diagram.

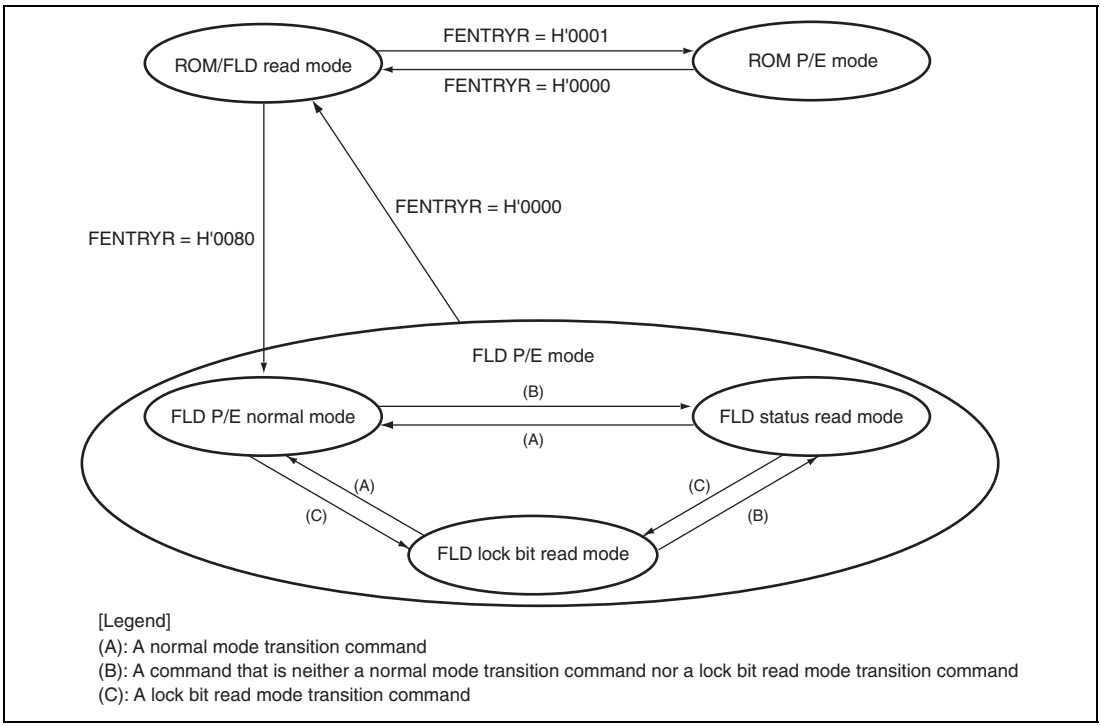


Figure 24.5 FCU Mode Transition Diagram (FLD-Related Modes)

(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 23.6.2, 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 24.8 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 24.8 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 24.8 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 24.8 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 24.7.3, 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 24.8, 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 24.8 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 24.8 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

24.6.3 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 23.6.3, 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 29, 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

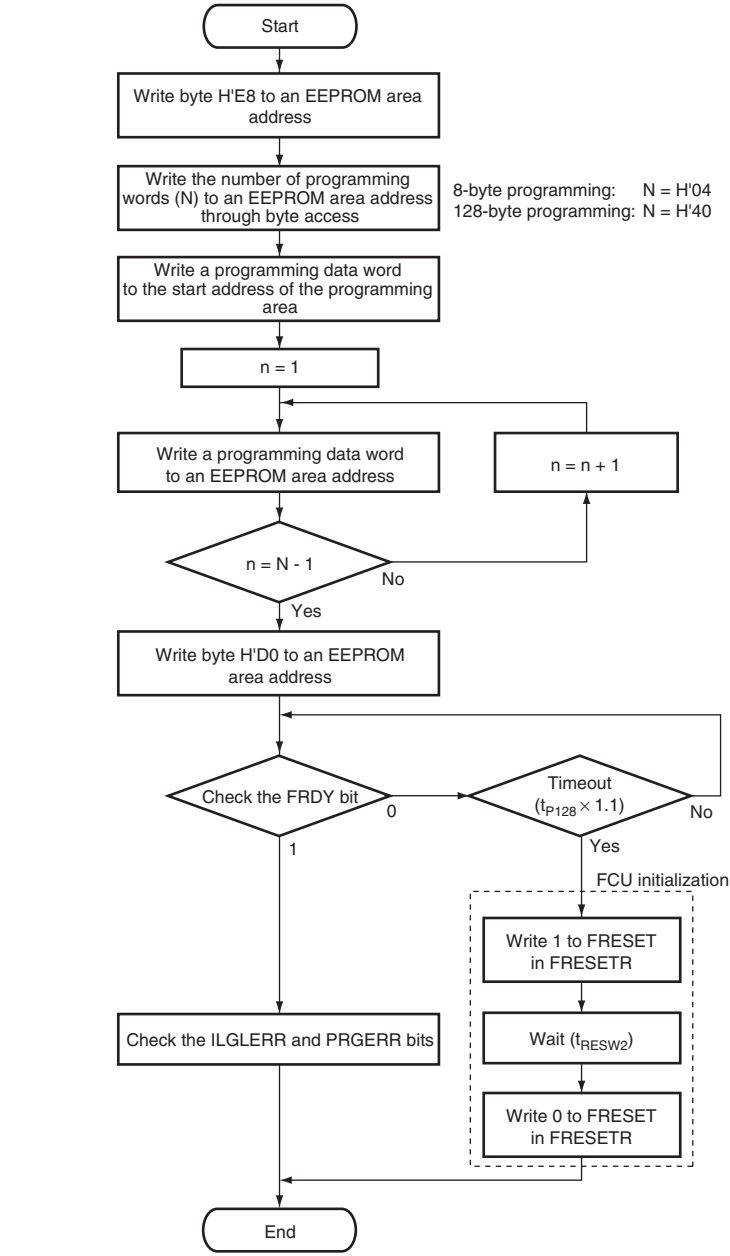
The command is used for notification of the peripheral clock frequency. For details, see section 23.6.3, FCU Command Usage, in section 23, Flash Memory (ROM). Proceed by setting the FENTRYD bit in FENTRYR to 1 and specifying the address as an address within the region corresponding to the data flash (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 24.6 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 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 24.6 Procedure for FLD Programming

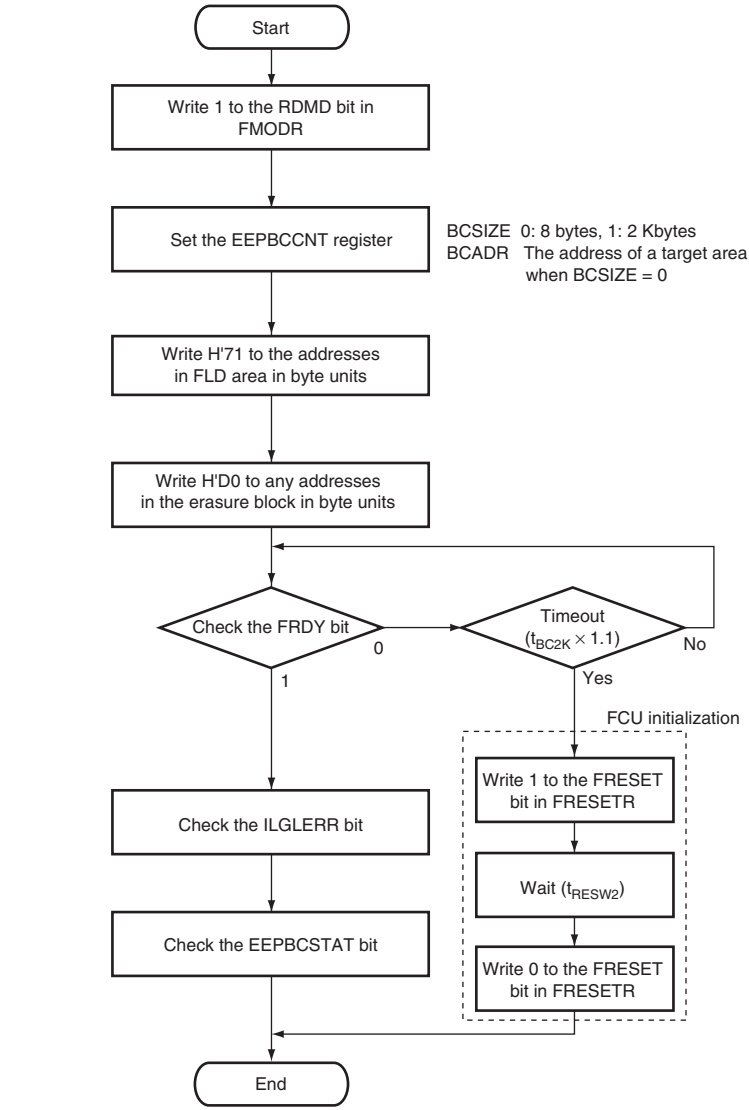
(3) Erasure

To erase the ROM, use the block erase command. The FLD can be erased in the same way as ROM erasure (refer to section 23, Flash Memory (ROM)). Note that the FLD has a programming and erasure protection function through EEPWE0 and EEPWE1. 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 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 (2 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 24.7 shows the procedure of the FLD blank check.



Notes: t_{BC2K}: Time required for blank check (see section 29, Electrical Characteristics).
t_{RESW2}: Reset pulse width during programming and erasure (see section 29, Electrical Characteristics).

Figure 24.7 Procedure of the FLD Blank Check

24.7 Protection

There are three types of FLD programming/erasure protection: hardware, software, and error protection.

24.7.1 Hardware Protection

The hardware protection function disables FLD programming and erasure according to the mode pin settings in this LSI.

For the operating modes set through the mode pins of this LSI, refer to section 3, MCU Operating Modes.

24.7.2 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 24.7.3, Error Protection).

(2) Protection through EEPWE0

When the DBWEi (i = 00 to 15) bit in EEPWE0 or EEPWE1 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 24.7.3, Error Protection).

24.7.3 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 24.9 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 23.9.3, 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 24.9 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 and EEPRE1.	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 and EEPWE1.	1	0	0	0	0	0	1

24.8 Usage Notes

24.8.1 Protection of Data MAT Immediately after a Reset

As the initial values of EEPRE0, EEPRE1, EEPWE0, and EEPWE1 are H'0000, data MAT programming, erasure, and reading are disabled immediately after a reset. To read data from the data MAT, set EEPRE0 and EEPRE1 appropriately before accessing the data MAT. To program or erase the data MAT, set EEPWE0 and EEPWE1 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.

24.8.2 State in which Interrupts are Ignored

In the following modes or period, the NMI or maskable interrupt requests are ignored.

- 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

24.8.3 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.

24.8.4 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.

24.8.5 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 29, 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 29, 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.

24.8.6 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.

24.8.7 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.

24.8.8 Program for Reading

Execute program code for reading the FLD from on-chip RAM or on-chip ROM.

24.8.9 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

Section 25 On-Chip RAM

SH7239 and SH7237 Groups incorporate 64-Kbyte or 32-Kbyte RAM. For the capacity of the RAM in each product, see section 1.2, List of Products. The RAM 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 25.1 shows a block diagram of the RAM and figure 25.2 shows RAM and bus connections.

25.1 Features

- Page

  Pages 0 and 1: SH72394, SH72374

  Pages 0, 1, 4, and 5: SH72395, SH72375
- 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.

  Pages 0 and 1: One cycle in case of writing and reading

  Pages 4, and 5: 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).

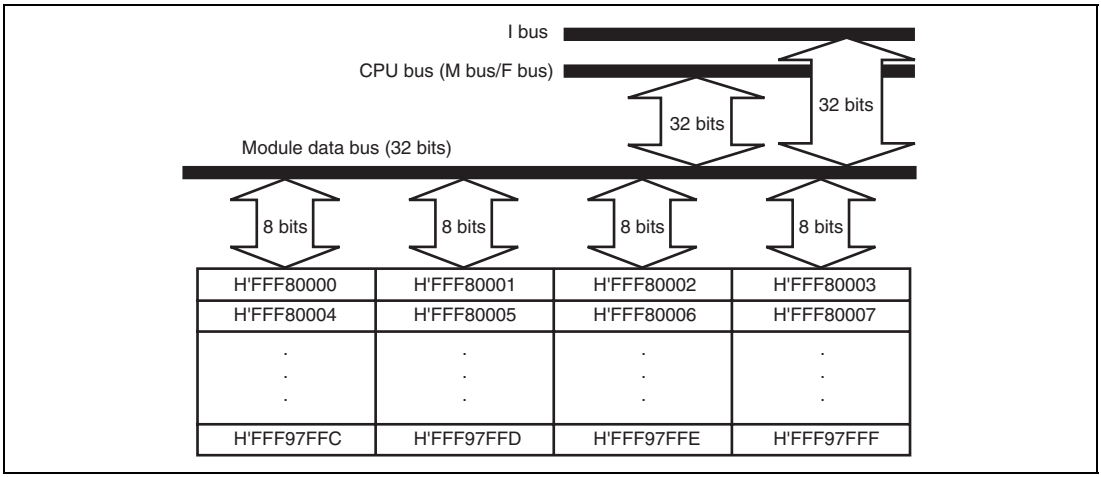


Figure 25.1 RAM Block Diagram

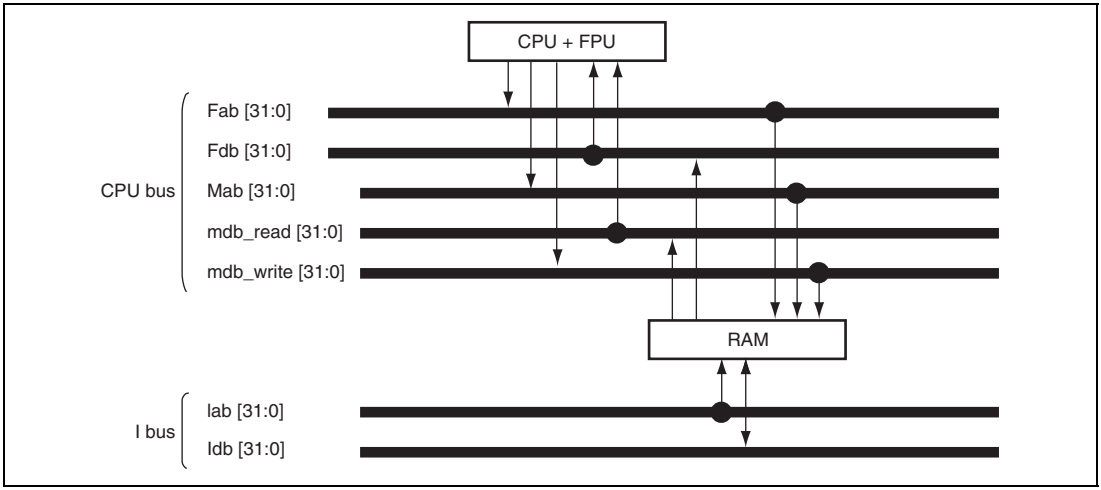


Figure 25.2 Bus Connections in RAM

Table 25.1 On-chip RAM Address Space

Page	Address
Page 0	H'FFF80000 to H'FFF83FFF
Page 1	H'FFF84000 to H'FFF87FFF
Page 4	H'FFF90000 to H'FFF93FFF
Page 5	H'FFF94000 to H'FFF97FFF

Note: The available pages differ depending on the product.
Pages 0 and 1: SH72394, SH72374
Pages 0, 1, 4, and 5: SH72395, SH72375

25.2 Register Descriptions

The on-chip RAM has registers shown in table 25.2.

Table 25.2 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

25.2.1 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
	—	—	RAME5	RAME4	—	—	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	RAME5	1	R/W	RAM Enable 5 Enables or disables accesses to page 5. 0: Accesses to page 5 disabled 1: Accesses to page 5 enabled
4	RAME4	1	R/W	RAM Enable 4 Enables or disables accesses to page 4. 0: Accesses to page 4 disabled 1: Accesses to page 4 enabled
3, 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	RAME1	1	R/W	RAM Enable 1 Enables or disables accesses to page 1. 0: Accesses to page 1 disabled 1: Accesses to page 1 enabled
0	RAME0	1	R/W	RAM Enable 0 Enables or disables accesses to page 0. 0: Accesses to page 0 disabled 1: Accesses to page 0 enabled

Note: The RAME4 and RAME5 bits are reserved in the SH72394 and SH72374.

25.2.2 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
	—	—	RAM WE5	RAM WE4	—	—	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Descriptions
7, 6	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
5	RAMWE5	1	R/W	RAM Write Enable 5 Enables or disables write to page 5. 0: Write to page 5 disabled 1: Write to page 5 enabled
4	RAMWE4	1	R/W	RAM Write Enable 4 Enables or disables write to page 4. 0: Write to page 4 disabled 1: Write to page 4 enabled
3, 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	RAMWE1	1	R/W	RAM Write Enable 1 Enables or disables write to page 1. 0: Write to page 1 disabled 1: Write to page 1 enabled
0	RAMWE0	1	R/W	RAM Write Enable 0 Enables or disables write to page 0. 0: Write to page 0 disabled 1: Write to page 0 enabled

Note: The RAMWE4 and RAMWE5 bits are reserved in the SH72394 and SH72374.

25.3 Notes on Usage

25.3.1 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 26 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.

26.1 Features

26.1.1 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 26.1 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 26.1 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, DMAC, DTC, and flash memory)

Note: * The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

26.1.2 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 RSTE 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 27, 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 RSTE bit in WRCSR is 1, and the counter overflows.

26.2 Input/Output Pins

Table 26.2 lists the pins used for power-down modes.

Table 26.2 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.

26.3 Register Descriptions

The following registers are used in power-down modes.

Table 26.3 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

26.3.1 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.

26.3.2 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.

26.3.3 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 32	-	MSTP 30
Initial value:	0	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R	R	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, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
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.

26.3.4 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	-	-
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, 0	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

26.3.5 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	-	-	MSTP52	MSTP51	MSTP50
Initial value:	1	1	1	1	1	1	1	1
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	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, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
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	MSTP51	1	R/W	Module Stop 51 When the MSTP51 bit is set to 1, the supply of the clock to the ADC2 is halted. 0: ADC2 runs. 1: Clock supply to ADC12 halted.
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.

26.3.6 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
	-	-	-	MSTP64	-	-	-	-
Initial value:	1	1	0	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R	R

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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
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.

26.4 Operation

26.4.1 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.

26.4.2 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 ROM cache remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. Table 26.4 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 26.4 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
Serial communication interface (SCI)	—	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 (IRQ6 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.

26.4.3 Application Example of Software Standy Mode

Figure 26.1 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.

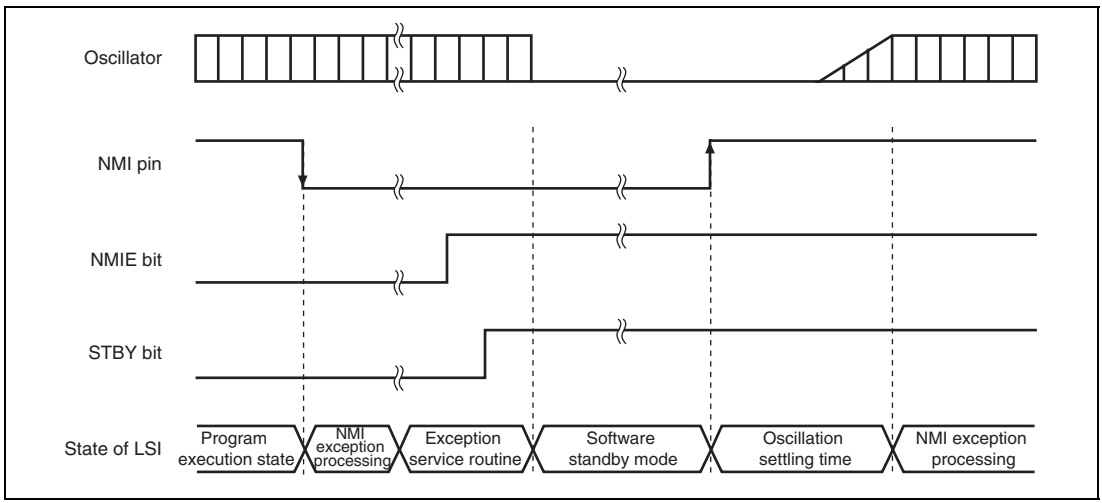


Figure 26.1 NMI Timing in Software Standby Mode (Application Example)

26.4.4 Module Standby Function

(1) Transition to Module Standby Function

Setting the MSTP bits in standby control registers (STBCR2 to STBCR6) 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, DTC, and flash memory). 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.

Section 27 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for emulator support.

27.1 Features

The user debugging interface (H-UDI) has reset and interrupt request functions.

The H-UDI in this LSI is used for emulator connection. Refer to the emulator manual for the method of connecting the emulator.

Figure 27.1 shows a block diagram of the H-UDI.

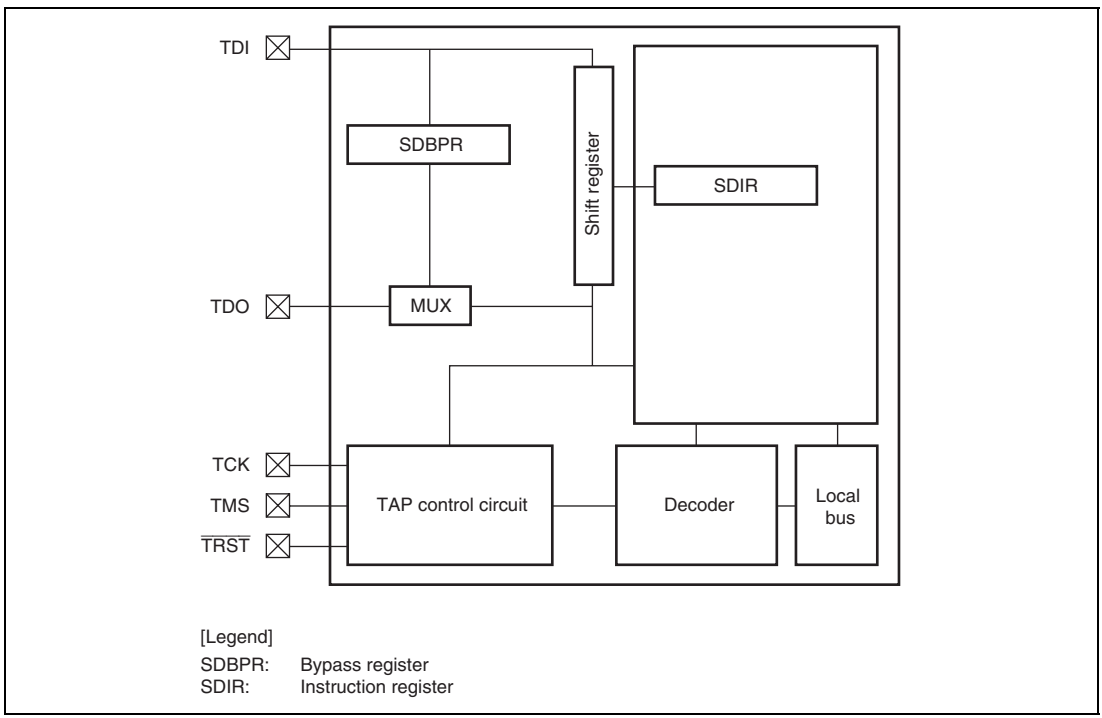


Figure 27.1 Block Diagram of H-UDI

27.2 Input/Output Pins

Table 27.1 Pin Configuration

Pin Name	Symbol	I/O	Function
H-UDI 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.
Mode select input pin	TMS	Input	The state of the TAP control circuit is determined by changing this signal in synchronization with TCK. For the protocol, see figure 27.2.
H-UDI reset input pin	$\overline{\text{TRST}}$	Input	Input is accepted asynchronously with respect to TCK, and when low, the H-UDI is reset. $\overline{\text{TRST}}$ must be low for a constant period when power is turned on regardless of using the H-UDI function. See section 27.4.2, Reset Configuration, for more information.
H-UDI serial data input pin	TDI	Input	Data transfer to the H-UDI is executed by changing this signal in synchronization with TCK.
H-UDI serial data output pin	TDO	Output	Data read from the H-UDI is executed by reading this pin in synchronization with TCK. The initial value of the data output timing is the TCK falling edge. This can be changed to the TCK rising edge by inputting the TDO change timing switch command to SDIR. See section 27.4.3, TDO Output Timing, for more information.
ASE mode select 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, normal mode is entered. In ASE mode, dedicated emulator function can be used. The input level at the $\overline{\text{ASEMD0}}$ pin should be held for at least one cycle after $\overline{\text{RES}}$ negation.

Note: * When the emulator is not in use, fix this pin to the high level.

27.3 Register Descriptions

The H-UDI has the following registers.

Table 27.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	SDBPR	—	—	—	—
Instruction register	SDIR	R	H'EFFD	H'FFFE2000	16

27.3.1 Bypass Register (SDBPR)

SDBPR is a 1-bit register that cannot be accessed by the CPU. When SDIR is set to BYPASS mode, SDBPR is connected between H-UDI pins TDI and TDO. The initial value is undefined.

27.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. It is initialized by $\overline{\text{TRST}}$ assertion or in the TAP test-logic-reset state, and can be written to by the H-UDI irrespective of CPU mode. Operation is not guaranteed if a reserved command is set in this register. The initial value is H'EFFD.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI[7:0]								-	-	-	-	-	-	-	-
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

Note: * The initial value of the TI[7:0] bits is a reserved value. When setting a command, the TI[7:0] bits must be set to another value.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	TI[7:0]	11101111*	R	Test Instruction The H-UDI instruction is transferred to SDIR by a serial input from TDI. For commands, see table 27.3.
7 to 2	—	All 1	R	Reserved These bits are always read as 1.
1	—	0	R	Reserved This bit is always read as 0.
0	—	1	R	Reserved This bit is always read as 1.

Table 27.3 H-UDI Commands

Bits 15 to 8								
TI7	**TI6**	**TI5**	**TI4**	**TI3**	**TI2**	**TI1**	**TI0**	**Description**
0	1	1	0	—	—	—	—	H-UDI reset negate
0	1	1	1	—	—	—	—	H-UDI reset assert
1	0	0	1	1	1	0	0	TDO change timing switch
1	0	1	1	—	—	—	—	H-UDI interrupt
1	1	1	1	—	—	—	—	BYPASS mode
Other than above								Reserved

27.4 Operation

27.4.1 TAP Controller

Figure 27.2 shows the internal states of the TAP controller.

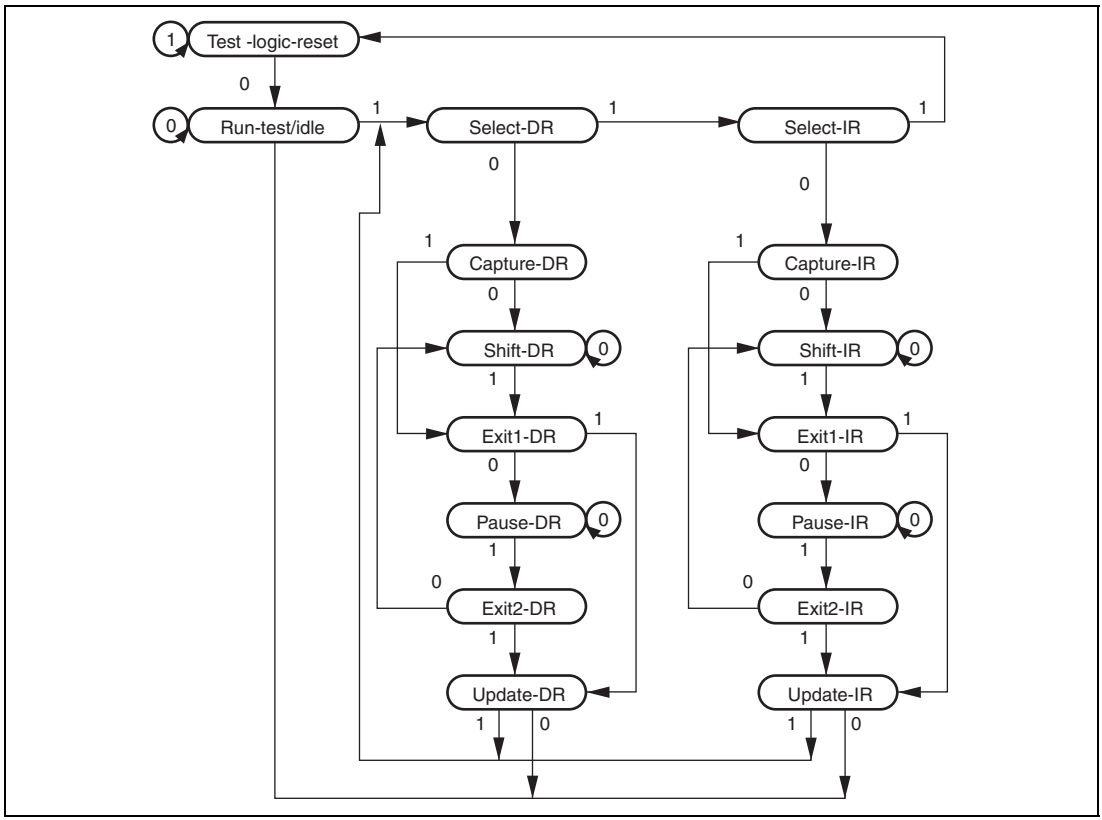


Figure 27.2 TAP Controller State Transitions

Note: The transition condition is the TMS value at the rising edge of TCK. The TDI value is sampled at the rising edge of TCK; shifting occurs at the falling edge of TCK. For details on change timing of the TDO value, see section 27.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. There is a transition to test-logic-reset asynchronously with TCK due to $\overline{\text{TRST}}$ assertion.

27.4.2 Reset Configuration

Table 27.4 Reset Configuration

$\overline{\text{ASEMD0}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	Chip State
H	L	L	Power-on reset and H-UDI reset
		H	Power-on reset
	H	L	H-UDI reset only (Normal operation)
		H	Normal operation
L	L	L	Reset hold*2
		H	Power-on reset
	H	L	H-UDI reset only
		H	Normal operation

Notes: 1. Performs normal mode and ASE mode settings
$\overline{\text{ASEMD0}}$ = H, normal mode
$\overline{\text{ASEMD0}}$ = L, ASE mode
2. In ASE mode, reset hold is entered if the $\overline{\text{TRST}}$ pin is driven low while the $\overline{\text{RES}}$ pin is negated. In this state, the CPU does not start up. When $\overline{\text{TRST}}$ is driven high, H-UDI operation is enabled, but the CPU does not start up. The reset hold state is cancelled by a power-on reset.

27.4.3 TDO Output Timing

The initial value of the TDO change timing is to perform data output from the TDO pin on the TCK falling edge. However, setting a TDO change timing switch command in SDIR via the H-UDI pin and passing the Update-IR state synchronizes the TDO change timing to the TCK rising edge. Thereafter the TDO change timing cannot be changed unless a power-on reset that asserts the $\overline{\text{TRST}}$ pin simultaneously is performed.

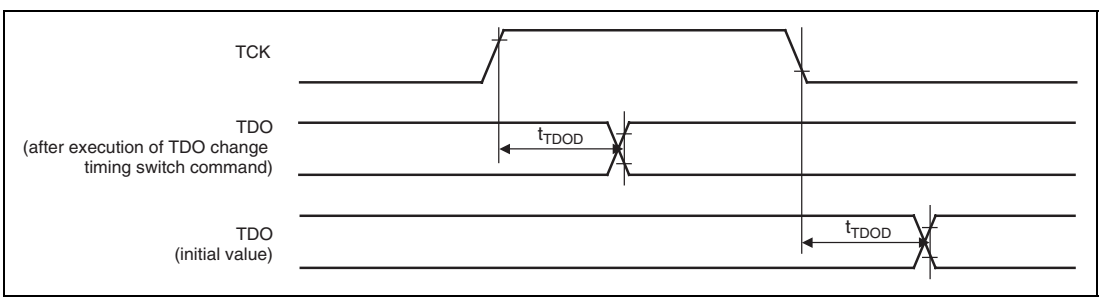


Figure 27.3 H-UDI Data Transfer Timing

27.4.4 H-UDI Reset

An H-UDI reset is executed by setting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by setting an 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 time for keeping the $\overline{\text{RES}}$ pin low to apply a power-on reset.

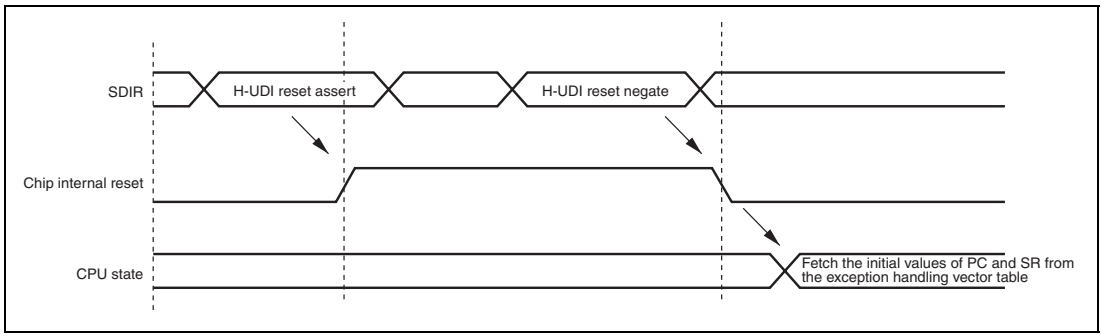


Figure 27.4 H-UDI Reset

27.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in fetching the exception service routine start address from the exception handling vector table, jumping to that address, and starting program execution from that address. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in software standby mode.

27.5 Usage Notes

1. An H-UDI command, once set, will not be modified as long as another command is not set again from the H-UDI. If the same command is to be set continuously, the command must be set after a command (BYPASS mode, etc.) that does not affect chip operations is once set.
2. In software standby mode, this LSI stops operation and does not accept any H-UDI command. To retain the TAP status before and after software standby mode, keep TCK high before entering software standby mode.

Section 28 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.

28.1 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
	MTU 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

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
	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
	CS0 space wait control register	CS0WCR	32	H'FFFC0028	32
	CS1 space wait control register	CS1WCR	32	H'FFFC002C	32
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
BSC	CS6 space wait control register	CS6WCR	32	H'FFFC0040	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
	Port output enable control register 3	POECR3	8	H'FFFE500E	8
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	16*[1]
	Watchdog timer counter	WTCNT	16	H'FFFE0002	16*[1]
	Watchdog reset control/status register	WRCSR	16	H'FFFE0004	16*[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
	Sampling mode register_0	SPMR_0	8	H'FFFF8014	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
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
	Sampling mode register_1	SPMR_1	8	H'FFFF8814	8
SCI (channel 2)	Serial mode register_2	SCSMR_2	8	H'FFFF9000	8
	Bit rate register_2	SCBRR_2	8	H'FFFF9002	8
	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
	Sampling mode register_2	SPMR_2	8	H'FFFF9014	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
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
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
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
	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
	A/D control register_2	ADCR_2	8	H'FFFFEE00	8
	A/D status register_2	ADSR_2	8	H'FFFFEE02	8
	A/D start trigger select register_2	ADSTRGR_2	8	H'FFFFEE1C	8
	A/D analog input channel select register_2	ADANSR_2	8	H'FFFFEE20	8
	A/D bypass control register_2	ADBYPSCR_2	8	H'FFFFEE30	8
	A/D trigger select register_0	ADTSR_0	16	H'FFFFE930	16
	A/D trigger select register_1	ADTSR_1	16	H'FFFFED30	16
	A/D trigger select register_2	ADTSR_2	16	H'FFFFEF30	16
	A/D data register 8	ADDR8	16	H'FFFFEE40	16
	A/D data register 9	ADDR9	16	H'FFFFEE42	16
	A/D data register 10	ADDR10	16	H'FFFFEE44	16
	A/D data register 11	ADDR11	16	H'FFFFEE46	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC	A/D data register 12	ADDR12	16	H'FFFFEE48	16
	A/D data register 13	ADDR13	16	H'FFFFEE4A	16
	A/D data register 14	ADDR14	16	H'FFFFEE4D	16
	A/D data register 15	ADDR15	16	H'FFFFEE4E	16
	A/D group-0 data-0 register_0	ADDR0GR0_0	16	H'FFFFE932	16
	A/D group-0 data-0 register_1	ADDR0GR0_1	16	H'FFFFED32	16
	A/D group-0 data-0 register_2	ADDR0GR0_2	16	H'FFFFEF32	16
	A/D group-1 data-2 register_0	ADDR2GR1_0	16	H'FFFFE934	16
	A/D group-1 data-2 register_1	ADDR2GR1_1	16	H'FFFFED34	16
	A/D group-1 data-2 register_2	ADDR2GR1_2	16	H'FFFFEF34	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 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 H	PBIORH	16	H'FFFE3884	8, 16, 32
	Port B IO register L	PBIORL	16	H'FFFE3886	8, 16
	Port B control register H2	PBCRH2	16	H'FFFE388C	8, 16, 32
	Port B control register H1	PBCRH1	16	H'FFFE388E	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 H	PBPCRH	16	H'FFFE38A8	8, 16, 32
	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 L	PDIORL	16	H'FFFE3986	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
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	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
	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 H	PBDRH	16	H'FFFE3880	8, 16, 32
	Port B data register L	PBDRL	16	H'FFFE3882	8, 16
	Port B port register H	PBPRH	16	H'FFFE389C	8, 16, 32
	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 L	PDDRL	16	H'FFFE3982	8, 16
	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
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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ROM/FLD	FCU command register	FCMDR	16	H'FFFFA90A	8, 16
	FCU processing switch register	FCPSR	16	H'FFFFA918	8, 16
	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
	Peripheral clock notification register	PCKAR	16	H'FFFFA938	8, 16
	FLD read enable register 0	EEPRE0	16	H'FFFFA840	8, 16
	FLD read enable register 1	EEPRE1	16	H'FFFFA842	8, 16
	FLD program/erase enable register 0	EEPWE0	16	H'FFFFA850	8, 16
	FLD program/erase enable register 1	EEPWE1	16	H'FFFFA852	8, 16
	ROM cache control register	RCCR	32	H'FFFC1400	32
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.

28.2 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	—	—	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	IRQRR	—	—	—	—	—	—	—	—
		—	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				—	—	—	—
	IPR05	—	—	—	—	—	—	—	—
		ADI0				ADI1			
	IPR06	DMAC0				DMAC1			
		DMAC2				DMAC3			
	IPR07	DMAC4				DMAC5			
		DMAC6				DMAC7			
	IPR08	CMT0				CMT1			
		—	—	—	—	WDT			
	IPR09	MTU2_0				MTU2_0			
		MTU2_1				MTU2_1			
	IPR10	MTU2_2				MTU2_2			
		MTU2_3				MTU2_3			
	IPR11	MTU2_4				MTU2_4			
		MTU2_5				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	MTU2S_3				MTU2S_3			
		MTU2S_4				MTU2S_4			
	IPR13	MTU2S_5				POE2			
		—	—	—	—	—	—	—	—
	IPR14	—	—	—	—	—	—	—	—
		—	—	—	—	SCIF3			
	IPR15	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	IPR16	SCI0				SCI1			
		SCI2				—	—	—	—
	IPR17	RSPI				—	—	—	—
		ADI2				—	—	—	—
	IPR18	—	—	—	—	RCAN-ET			
		—	—	—	—	—	—	—	—
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	—
		DTCERA7	DTCERA6	DTCERA5	DTCERA4	DTCERA3	DTCERA2	—	—
	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
		—	—	DTCERD5	DTCERD4	—	—	—	—

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
		—	—	—	—	—	—	—	—
	DTCCR	—	—	—	RRS	RCHNE	—	—	ERR
	DTCVBR								
						—	—	—	—
		—	—	—	—	—	—	—	—
BSC	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY[1:0]		DMAIW[2]
		DMAIW[1:0]		DMAIWA	—	—	HIZCKIO	HIZMEM	—
	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]		—
		—	—	—	—	—	—	—	—
	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]		—
		—	—	—	—	—	—	—	—
	CS0WCR*	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS1WCR*	—	—	—	—	—	—	—	—
		—	—	—	—	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS3WCR*	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	WR[3:1]		
		WR[0]	WM	—	—	—	—	—	—
	CS4WCR*	—	—	—	—	—	—	—	—
		—	—	—	—	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS5WCR*	—	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS6WCR*	—	—	—	—	—	—	—	—
		—	—	—	—	—	—		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	BSCEHR	DTLOCK	—	—	—	DTBST	DTSA	—	DTPR
		—	—	—	—	—	—	—	—

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_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
	RSAR_0								
	RDAR_0								
	RDMATCR_0								
	SAR_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	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								
	RDMATCR_1								
	SAR_2								
	DAR_2								
	DMATCR_2								

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_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								
	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								

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_3								
	RDMATCR_3								
	SAR_4								
	DAR_4								
	DMATCR_4								
	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								

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_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								
	RDAR_5								
	RDMATCR_5								
	SAR_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	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								
	SAR_7								
	DAR_7								

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	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]	
	DMARS3	CH7MID[5:0]						CH7RID[1:0]	
		CH6MID[5:0]						CH6RID[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
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	—	—	—	—	—	—	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	—	—	—	—	—	—
	TSYRS	SYNC4	SYNC3	—	—	—	—	—	—
	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	—	—	—	POE0F	—	—	—	PIE1
		—	—	—	—	—	—	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	—	—	—	—
	POECR3	—	—	IC2MTU2 CH0ZE	IC2MTU2 ZE	IC3MTU2S ZE	IC3MTU2 CH0ZE	IC1MTU2 CH0ZE	IC1MTU2S ZE
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
	SPMR_0	—	—	—	—	—	—	—	STDSPM
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
	SPMR_1	—	—	—	—	—	—	—	STDSPM
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
	SPMR_2	—	—	—	—	—	—	—	STDSPM
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	—	—	—	—	—	—	SPSLN[1:0]	
	SPSSR	—	—	SPECM[1:0]		—	—	SPCP[1:0]	
	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
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
	ADTSR_0	TRG1S[3:0]				TRG0S[3:0]			
		—	—	—	—	—	CHSEC	CONADF	2CHSE
	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
	ADDR0GR0_0	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR2GR1_0	—	—	—	—	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	—	—	—	—	—	—	—	—
	ADTSR_1	TRG1S[3:0]				TRG0S[3:0]			
		—	—	—	—	—	CHSEC	CONADF	2CHSE
	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
ADC	ADDR0GR0_1	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR2GR1_1	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADCR_2	ADST	ADCS	ACE	ADIE	—	—	TRGE	EXTRG
	ADSR_2	—	—	—	—	—	—	—	ADF
	ADSTRGR_2	—	STR6	STR5	STR4	STR3	STR2	STR1	STR0
	ADANSR_2	—	—	—	—	ANS3	ANS2	ANS1	ANS0
	ADBYPSCR_2	—	—	—	—	ADSST	—	—	—
	ADTSR_2	TRG1S[3:0]				TRG0S[3:0]			
		—	—	—	—	—	CHSEC	CONADF	2CHSE
	ADDR8	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR9	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR10	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR11	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR12	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR13	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR14	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR15	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR0GR0_2	—	—	—	—	ADD11	ADD10	ADD9	ADD8
		ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
	ADDR2GR1_2	—	—	—	—	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	—	—	—	—	—	—	—	—
		—	—	—	—	—	PA18IOR	PA17IOR	PA16IOR
	PAIORL	PA15IOR	—	—	—	—	—	PA9IOR	PA8IOR
		PA7IOR	PA6IOR	—	—	—	—	PA1IOR	PA0IOR
	PACRH1	—	—	—	—	—	PA18MD[2:0]		
		—	PA17MD[2:0]			—	PA16MD[2:0]		
	PACRL4	—	PA15MD[2:0]			—	—	—	—
		—	—	—	—	—	—	—	—
	PACRL3	—	—	—	—	—	—	—	—
		—	PA9MD[2:0]			—	PA8MD[2:0]		
	PACRL2	—	PA7MD[2:0]			—	PA6MD[2:0]		
		—	—	—	—	—	—	—	—
	PACRL1	—	—	—	—	—	—	—	—
		—	PA1MD[2:0]			—	PA0MD[2:0]		
	PAPCRH	—	—	—	—	—	—	—	—
		—	—	—	—	—	PA18PCR	PA17PCR	PA16PCR
	PAPCRL	PA15PCR	—	—	—	—	—	PA9PCR	PA8PCR
		PA7PCR	PA6PCR	—	—	—	—	PA1PCR	PA0PCR
	PBIORH	—	—	—	—	—	—	—	—
		—	—	PB21IOR	PB20IOR	PB19IOR	PB21IOR	PB18IOR	PB16IOR

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	—	—	—	—	—	—	—	—
		—	—	—	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
	PBCRH2	—	—	—	—	—	—	—	—
		—	PB21MD[2:0]			—	PB20MD[2:0]		
	PBCRH1	—	PB19MD[2:0]			—	PB18MD[2:0]		
		—	PB17MD[2:0]			—	PB16MD[2:0]		
	PBCRL2	—	—	—	—	—	—	—	—
		—	—	—	—	—	PB4MD[2:0]		
	PBCRL1	—	PB3MD[2:0]			—	PB2MD[2:0]		
		—	PB1MD[2:0]			—	PB0MD[2:0]		
	PBPCRH	—	—	—	—	—	—	—	—
		—	—	PB21PCR	PB20PCR	PB19PCR	PB21PCR	PB18PCR	PB16PCR
	PBPCRL	—	—	—	—	—	—	—	—
		—	—	—	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
	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR

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	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]		
	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	—	—	—	—	—	—	—	—
		—	—	—	—	—	MZIZDL	MZIZEH	MZIZEL
	PDACKCR	—	—	—	—	—	—	—	—
		—	—	—	—	DACK3TMG	DACK2TMG	DACK1TMG	DACK0TMG

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	PEPCRL	PE15PCR	PE14PCR	PE13PCR	PE12PCR	PE11PCR	PE10PCR	PE9PCR	PE8PCR
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
I/O port	PADRH	—	—	—	—	—	—	—	—
		—	—	—	—	—	PA18DR	PA17DR	PA16DR
	PADRL	PA15DR	—	—	—	—	—	PA9DR	PA8DR
		PA7DR	PA6DR	—	—	—	—	PA1DR	PA0DR
	PAPRH	—	—	—	—	—	—	—	—
		—	—	—	—	—	PA18PR	PA17PR	PA16PR
	PAPRL	PA15PR	—	—	—	—	—	PA9PR	PA8PR
		PA7PR	PA6PR	—	—	—	—	PA1PR	PA0PR
	PBDRH	—	—	—	—	—	—	—	—
		—	—	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR
	PBDRL	—	—	—	—	—	—	—	—
		—	—	—	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPRH	—	—	—	—	—	—	—	—
		—	—	PB21PR	PB20PR	PB19PR	PB18PR	PB17PR	PB16PR
	PBPRL	—	—	—	—	—	—	—	—
		—	—	—	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
	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
	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	PF15PR	PF14PR	PF13PR	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR
		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
ROM/FLD	FPMON	FWE	—	—	—	—	—	—	—
	FMODR	—	—	—	FRDMD	—	—	—	—
	FASTAT	ROMAE	—	—	CMDLK	EEPAE	EEPIFE	EEPRPE	EEPWPE
	FAEINT	ROMAEIE	—	—	CMDLKIE	EEPAEIE	EEPIFEIE	EEPRPEIE	EEPWPEIE
	ROMMAT	KEY							
		—	—	—	—	—	—	—	ROMSEL
	FCURAME	KEY							
		—	—	—	—	—	—	—	FCRME
	FSTATR0	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
	FSTATR1	FCUERR	—	—	FLOCKST	—	—	—	—
	FENTRYR	FKEY							
		FENTRYD	—	—	—	—	—	—	FENTRY0
	FPROTR	FPKEY							
		—	—	—	—	—	—	—	FPROTCN
	FRESETR	FPKEY							
		—	—	—	—	—	—	—	FRESET
	FCMDR	CMDR							
		PCMDR							
	FCPSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ESUSPMD
	EEPBCCNT	—	—	—	BCADR				
		BCADR					—	—	BCSIZE
	FPESTAT	—	—	—	—	—	—	—	—
		PEERRST							
	EEPBCSTAT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	BCST
	PCKAR	—	—	—	—	—	—	—	—
		PCKA							
	EEPRE0	KEY							
		DBRE07	DBRE06	DBRE05	DBRE04	DBRE03	DBRE02	DBRE01	DBRE00
	EEPRE1	KEY							
		DBRE15	DBRE14	DBRE13	DBRE12	DBRE11	DBRE10	DBRE09	DBRE08
	EEPWE0	KEY							
		DBWE07	DBWE06	DBWE05	DBWE04	DBWE03	DBWE02	DBWE01	DBWE00
	EEPWE1	KEY							
		DBWE15	DBWE14	DBWE13	DBWE12	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	—	—	—
Power-down mode	STBCR	STBY	—	—	—	—	—	—	—
	STBCR2	MSTP10	MSTP9	MSTP8	—	—	—	MSTP4	—
	SYSCR1	—	—	RAME5	RAME4	—	—	RAME1	RAME0
	SYSCR2	—	—	RAMWE5	RAMWE4	—	—	RAMWE1	RAMWE0
	STBCR3	HIZ	MSTP36	MSTP35	—	—	MSTP32	—	MSTP30
	STBCR4	—	—	—	MSTP44	—	MSTP42	—	—
	STBCR5	MSTP57	MSTP56	MSTP55	—	—	MSTP52	MSTP51	MSTP50
	STBCR6	—	—	—	MSTP64	—	—	—	—
H-UDI	SDIR	TI[7:0]							
		—	—	—	—	—	—	—	—

Note: * When normal space or MPX-I/O is the memory type

28.3 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

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
	CS3BCR	Initialized	Retained	Retained	—	Retained
	CS4BCR	Initialized	Retained	Retained	—	Retained
	CS5BCR	Initialized	Retained	Retained	—	Retained
	CS6BCR	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
	CS3WCR	Initialized	Retained	Retained	—	Retained
	CS4WCR	Initialized	Retained	Retained	—	Retained
	CS5WCR	Initialized	Retained	Retained	—	Retained
	CS6WCR	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
	POECR3	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

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
SCI (channel 0)	SCSDCR_0	Initialized	Retained	Retained	Initialized	Retained
	SCSPTR_0	Initialized*[5]	Retained	Retained	Initialized	Retained
	SPMR_0	Initialized	Retained	Retained	Initialized	Retained
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
	SPMR_1	Initialized	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
	SPMR_2	Initialized	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

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
	ADTSR_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
	ADDR0GR0_0	Initialized	Retained	Initialized	Retained	Retained
	ADDR2GR1_0	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
	ADTSR_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
	ADDR0GR0_1	Initialized	Retained	Initialized	Retained	Retained
	ADDR2GR1_1	Initialized	Retained	Initialized	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
ADC	ADCR_2	Initialized	Retained	Initialized	Retained	Retained
	ADSR_2	Initialized	Retained	Initialized	Retained	Retained
	ADSTRGR_2	Initialized	Retained	Initialized	Retained	Retained
	ADANSR_2	Initialized	Retained	Initialized	Retained	Retained
	ADBYPSCR_2	Initialized	Retained	Initialized	Retained	Retained
	ADTSR_2	Initialized	Retained	Initialized	Retained	Retained
	ADDR8	Initialized	Retained	Initialized	Retained	Retained
	ADDR9	Initialized	Retained	Initialized	Retained	Retained
	ADDR10	Initialized	Retained	Initialized	Retained	Retained
	ADDR11	Initialized	Retained	Initialized	Retained	Retained
	ADDR12	Initialized	Retained	Initialized	Retained	Retained
	ADDR13	Initialized	Retained	Initialized	Retained	Retained
	ADDR14	Initialized	Retained	Initialized	Retained	Retained
	ADDR15	Initialized	Retained	Initialized	Retained	Retained
	ADDR0GR0_2	Initialized	Retained	Initialized	Retained	Retained
	ADDR2GR1_2	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	Initialized	Retained
	MB[0]. CONTROL1L	Initialized	Retained	Retained	Initialized	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
	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
	PBIORH	Initialized	Retained	Retained	—	Retained
	PBIORL	Initialized	Retained	Retained	—	Retained
	PBCRH2	Initialized	Retained	Retained	—	Retained
	PBCRH1	Initialized	Retained	Retained	—	Retained
	PBCRL2	Initialized	Retained	Retained	—	Retained
	PBCRL1	Initialized	Retained	Retained	—	Retained
	PBPCRH	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
	PDIORL	Initialized	Retained	Retained	—	Retained
	PDCRL4	Initialized	Retained	Retained	—	Retained
	PDCRL3	Initialized	Retained	Retained	—	Retained
	PDCRL2	Initialized	Retained	Retained	—	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
PFC	PDCRL1	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
	PDACKCR	Initialized	Retained	Retained	—	Retained
	PEPCRL	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
	PBDRH	Initialized	Retained	Retained	—	Retained
	PBDRL	Initialized	Retained	Retained	—	Retained
	PBPRH	—	Retained	Retained	—	Retained
	PBPRL	—	Retained	Retained	—	Retained
	PCDRL	Initialized	Retained	Retained	—	Retained
	PCPRL	—	Retained	Retained	—	Retained
	PDDRL	Initialized	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
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
	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
	PCKAR	Initialized	Retained	Retained	Retained	Retained
	EEPRE0	Initialized	Retained	Retained	Retained	Retained
	EEPRE1	Initialized	Retained	Retained	Retained	Retained
	EEPWE0	Initialized	Retained	Retained	Retained	Retained
	EEPWE1	Initialized	Retained	Retained	Retained	Retained
	RCCR	Initialized	Retained	Retained	Retained	Retained

Module Name	Register	Power-on Reset	Manual Reset	Software Standby	Module Standby	Sleep
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 29 Electrical Characteristics

29.1 Absolute Maximum Ratings

Table 29.1 lists the absolute maximum ratings.

Table 29.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage	SH7239B, SH7237B	VCC, PLLVCC	–0.3 to +7.0	V
	SH7239A, SH7237A	VCC, PLLVCC	–0.3 to +4.6	V
Input voltage (except analog input pins)		Vin	–0.3 to VCC +0.3	V
Analog power supply voltage		AVCC	–0.3 to +7.0	V
Analog reference voltage		AVREF	–0.3 to AVCC +0.3	V
Analog input voltage		V_{AN}	–0.3 to AVCC +0.3	V
Operating temperature	Industrial specifications	T_{opr}	–40 to +85	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note: Satisfy VCC ≤ AVCC. If not satisfied, current may flow.

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

29.2 DC Characteristics

Tables 29.2 to 29.6 list DC characteristics. Conditions for the indicated characteristics are as follows unless otherwise noted.

- SH7239A and SH7237A
  VCC = PLVCC = 3.0 to 3.6 V, AVCC = AVREF = 4.5 to 5.5 V,
  VSS = PLLVSS = AVREFVSS = AVSS = 0V, Ta = –40°C to +85°C
  (Industrial specifications)
- SH7239B and SH7237B
  VCC = PLVCC = 4.5 to 5.5 V, AVCC = AVREF = 4.5 to 5.5 V,
  VSS = PLLVSS = AVREFVSS = AVSS = 0V, Ta = –40°C to +85°C
  (Industrial specifications)

Table 29.2 DC Characteristics (SH7239A and SH7237A)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		VCC, PLLVCC	3.0	3.3	3.6	V	
Analog power supply voltage		AVCC	4.5	5.0	5.5	V	
Supply current*[1]	Normal operation	I_{CC}	—	90	120	mA	Iϕ = 160 MHz Bϕ = 40 MHz Pϕ = 40 MHz
	Increase in current during background operation (BGO)		—	20	—	mA	
	Software standby mode	I_{stby}	—	10	30	mA	VCC = 3.3 V
	Sleep mode	I_{sleep}	—	40	60	mA	VCC = 3.3 V
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1	µA	V_{in} = 0.5 to VCC – 0.5 V
Three-state leakage current	Input/output pins, all output pins (off state)	$\|I_{STI}\|$	—	—	1	µA	V_{in} = 0.5 to VCC – 0.5 V
Input capacitance	All pins	C_{in}	—	—	20	pF	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion	AI_{CC}	—	3	3.5	mA	Per 1 module
	Waiting for A/D conversion		—	30	50	µA	Per 1 module
	Standby		—	10	20	µA	Per 1 module
Reference power supply current	During A/D conversion	AIref	—	1	1.5	mA	Per 1 module
	Waiting for A/D conversion		—	1	1.5		Per 1 module
	Standby		—	0.8	5	µA	Per 1 module

Caution: When the A/D converter is not in use, the AVCC and AVSS pins should not be open. Connect the AVCC to the VCC.

Note: * Supply current values are when all output pins are unloaded. Other values for current do not include the increase in current while BGO is in use.

I_{CC}, I_{sleep}, and I_{stby} represent the total currents consumed in the VCC and PLLVCC systems.

Table 29.3 DC Characteristics (SH7239B and SH7237B)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		VCC, PLLVCC	4.5	5.0	5.5	V	
Analog power supply voltage		AVCC	4.5	5.0	5.5	V	
Supply current*[1]	Normal operation	I_{CC}	—	70	90	mA	Iϕ = 100 MHz Bϕ = 50 MHz Pϕ = 50 MHz
	Increase in current during background operation (BGO)		—	20	—	mA	
	Software standby mode	I_{stby}	—	10	30	mA	VCC = 5.0 V
	Sleep mode	I_{sleep}	—	40	60	mA	VCC = 5.0 V
Input leakage current	All input pins	$\|I_{in}\|$	—	—	1	µA	V_{in} = 0.5 to VCC – 0.5 V
Three-state leakage current	Input/output pins, all output pins (off state)	$\|I_{STI}\|$	—	—	1	µA	V_{in} = 0.5 to VCC – 0.5 V
Input capacitance	All pins	C_{in}	—	—	20	pF	

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion	AI_{CC}	—	3	3.5	mA	Per 1 module
	Waiting for A/D conversion		—	30	50	µA	Per 1 module
	Standby		—	10	20	µA	Per 1 module
Reference power supply current	During A/D conversion	AIref	—	1	1.5	mA	Per 1 module
	Waiting for A/D conversion		—	1	1.5		Per 1 module
	Standby		—	0.8	5	µA	Per 1 module

Caution: When the A/D converter is not in use, the AVCC and AVSS pins should not be open. Connect the AVCC to the VCC.

Note: * Supply current values are when all output pins are unloaded. Other values for current do not include the increase in current while BGO is in use.

I_{CC}, I_{sleep}, and I_{stby} represent the total currents consumed in the VCC and PLLVCC systems.

Table 29.4 DC Characteristics (SH7239A and SH7237A)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD0, FWE, $\overline{ASEMD0}$, $\overline{TRST}$, EXTAL	V_{IH}	VCC – 0.5	—	VCC + 0.3	V	VCC = 3.0 to 3.6 V
	Analog ports		2.2	—	AVCC + 0.3	V	AVCC = 3.0 to 5.5 V*
	Input pins other than above (excluding Schmitt pins)		2.2	—	VCC + 0.3		VCC = 3.0 to 3.6 V
Input low voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD0, FWE, $\overline{ASEMD0}$, $\overline{TRST}$, EXTAL	V_{IL}	–0.3	—	0.5	V	VCC = 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, $\overline{\text{POE8}}$, $\overline{\text{POE4}}$, $\overline{\text{POE0}}$, SCK3 to SCK0, RxD3 to RxD0, IRQ6 to IRQ0, RSPCK, MOSI, MISO, SSL0	V_T^+	VCC – 0.5	—	—	V	VCC = 3.0 to 3.6 V
		V_T^-	—	—	0.5	V	
		$V_T^+ - V_T^-$	VCC × 0.05	—	—	V	
Output high voltage	All output pins	V_{OH}	VCC – 0.5	—	—	V	I_{OH} = –200 µA
	TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS		VCC – 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, VCC = 3.0 to 3.6 V
	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	$-I_P$	–10	—	–800	µA	Vin = 0 V
	$\overline{\text{ASEMD0}}$						
RAM standby voltage		V_{RAM}	2.7	—	—	V	VCC

Note: * When the A/D converter is in use, AVCC must be from 4.5 to 5.5 V. When it is not in use, connect the AVCC to the VCC.

Table 29.5 DC Characteristics (SH7239B and SH7237B)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, MD0, FWE, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL	V_{IH}	VCC – 0.7	—	VCC + 0.3	V	VCC = 4.5 to 5.5 V
	Analog ports		2.2	—	AVCC + 0.3	V	AVCC = 4.5 to 5.5 V
	Input pins other than above (excluding Schmitt pins)		2.2	—	VCC + 0.3		VCC = 4.5 to 5.5 V
Input low voltage	$\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, MD0, FWE, $\overline{\text{ASEMD0}}$, $\overline{\text{TRST}}$, EXTAL	V_{IL}	–0.3	—	0.5	V	VCC = 4.5 to 5.5 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, $\overline{\text{POE8}}$, $\overline{\text{POE4}}$, $\overline{\text{POE0}}$, SCK3 to SCK0, RxD3 to RxD0, IRQ6 to IRQ0, RSPCK, MOSI, MISO, SSL0	V_T^+	VCC – 0.5	—	—	V	VCC = 4.5 to 5.5 V
		V_T^-	—	—	1.0	V	
		$V_T^+ - V_T^-$	0.2	—	—	V	
Output high voltage	All output pins	V_{OH}	VCC – 0.5	—	—	V	I_{OH} = –200 µA
	TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS		VCC – 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} = 15 mA, VCC = 4.5 to 5.5 V
	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	$-I_P$	–10	—	–800	µA	Vin = 0 V
	$\overline{\text{ASEMD0}}$						
RAM standby voltage		V_{RAM}	2.7	—	—	V	VCC

Table 29.6 Permissible Output Currents

Item	Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	I_{OL}	—	—	2.0*	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

Note: * TIOC3B, TIOC3D, TIOC4A to TIOC4D, TIOC3BS, TIOC3DS, TIOC4AS to TIOC4DS:
SH7239B and SH7237B; I_{OL} = 15 mA (Max.)/-I_{OH} = 5 mA (Max.),
SH7239A and SH7237A; I_{OL} = 10 mA (Max.)/-I_{OH} = 5 mA (Max.).
Of these pins, the number of pins from which current more than 2.0 mA runs evenly should be 3 or less.

Caution: To protect the LSI's reliability, do not exceed the output current values in table 29.6.

29.3 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.

Conditions for the indicated timings are as follows unless otherwise noted.

- SH7239A and SH7237A
  VCC = PLVCC = 3.0 to 3.6 V, AVCC = AVREF = 4.5 to 5.5 V,
  VSS = PLLVSS = AVREFVSS = AVSS = 0V, Ta = –40°C to +85°C
  (Industrial specifications)
- SH7239B and SH7237B
  VCC = PLVCC = 4.5 to 5.5 V, AVCC = AVREF = 4.5 to 5.5 V,
  VSS = PLLVSS = AVREFVSS = AVSS = 0V, Ta = –40°C to +85°C
  (Industrial specifications)

Table 29.7 Maximum Operating Frequency

Item		Symbol	Min.	Typ.	Max.	Unit
Operating frequency (SH7239A, SH7237A)	Internal clock (Iϕ)	f	40	—	160	MHz
	Bus clock (Bϕ)		20	—	40	
	Peripheral clock (Pϕ)		20	—	40	
	MTU clock (Mϕ)		40	—	80	
	AD clock (Aϕ)		40	—	40	
Operating frequency (SH7239B, SH7237B)	Internal clock (Iϕ)	f	40	—	100	MHz
	Bus clock (Bϕ)		20	—	50	
	Peripheral clock (Pϕ)		20	—	50	
	MTU clock (Mϕ)		40	—	100	
	AD clock (Aϕ)		40	—	50	

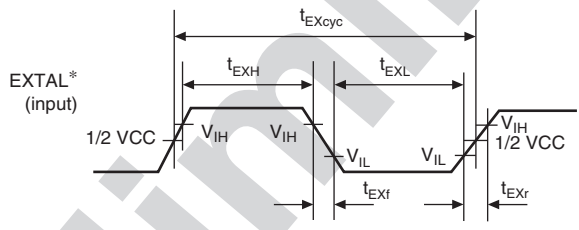
29.3.1 Clock Timing

Table 29.8 Clock Timing (SH7239A and SH7237A)

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	10	MHz	Figure 29.1
EXTAL clock input cycle time	t_{EXcyc}	100	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	40	MHz	Figure 29.2
CK clock output cycle time	t_{cyc}	25	50	ns	
CK clock output pulse low width	t_{CKOL}	6	—	ns	
CK clock output pulse high width	t_{CKOH}	6	—	ns	
CK clock output rise time	t_{CKOr}	—	3	ns	
CK clock output fall time	t_{CKOf}	—	3	ns	
Power-on oscillation setting time	t_{OSC1}	10	—	ms	Figure 29.3
Oscillation settling time on return from standby 1	t_{OSC2}	10	—	ms	Figure 29.4
Oscillation settling time on return from standby 2	t_{OSC3}	10	—	ms	Figure 29.5

Table 29.9 Clock Timing (SH7239B and SH7237B)

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	10	12.5	MHz	Figure 29.1
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	
Power-on oscillation setting time	t_{OSC1}	10	—	ms	Figure 29.3
Oscillation settling time on return from standby 1	t_{OSC2}	10	—	ms	Figure 29.4
Oscillation settling time on return from standby 2	t_{OSC3}	10	—	ms	Figure 29.5



Note: * When the clock is input on the EXTAL pin.

Figure 29.1 EXTAL Clock Input Timing

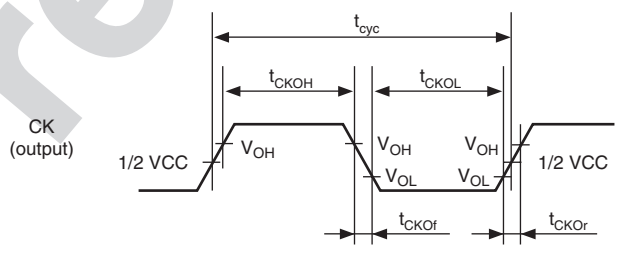
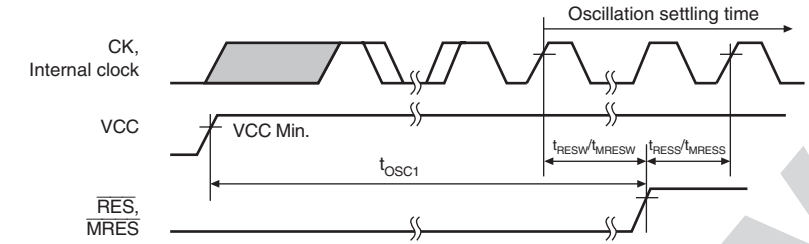
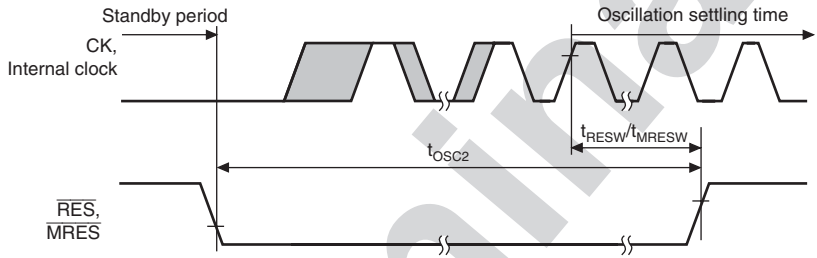


Figure 29.2 CK Clock Output Timing



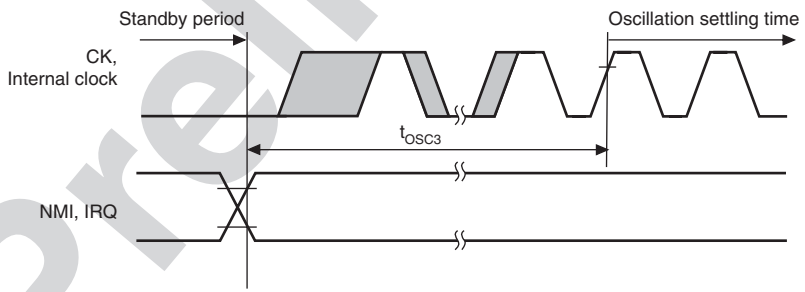
Note: Oscillation settling time when the internal oscillator is used.

Figure 29.3 Power-On Oscillation Settling Time



Note: Oscillation settling time when the internal oscillator is used.

Figure 29.4 Oscillation Settling Time on Return from Standby (Return by Reset)



Note: Oscillation settling time when the internal oscillator is used.

Figure 29.5 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

29.3.2 Control Signal Timing

Table 29.10 Control Signal Timing (SH7239A and SH7237A)

		Bϕ = 40 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{\text{RES}}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*[2]*[4]	—	t_{cyc}	Figures 29.3 to 29.6
		1.5*[4]	—	µs	
$\overline{\text{RES}}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100	—	µs	
$\overline{\text{RES}}$ setup time*[1]	t_{RESS}	100	—	ns	
$\overline{\text{RES}}$ hold time	t_{RESH}	15	—	ns	
$\overline{\text{MRES}}$ pulse width	t_{MRESW}	20*[3]	—	t_{cyc}	
$\overline{\text{MRES}}$ setup time	t_{MRESS}	120	—	ns	
$\overline{\text{MRES}}$ hold time	t_{MRESH}	15	—	ns	
MD0, FWE setup time	t_{MDS}	20	—	t_{cyc}	Figure 29.6
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	$1/2t_{cyc}$ + 15	—	ns	Figure 29.8
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 10	—	ns	
NMI setup time*[1]	t_{NMIS}	100	—	ns	Figure 29.7
NMI hold time	t_{NMIH}	10	—	ns	
IRQ6 to IRQ0 setup time*[1]	t_{IRQS}	35	—	ns	
IRQ6 to IRQ0 hold time	t_{IRQH}	10	—	ns	
IRQ pulse width	t_{IRQW}	4	—	t_{cyc}	
NMI pulse width	t_{NMIW}	4	—	t_{cyc}	
$\overline{\text{IRQOUT}}$ output delay time	t_{IRQOD}	—	100	ns	Figure 29.9
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	$1/2t_{cyc}$ + 20	ns	Figure 29.8
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. $\overline{\text{RES}}$, NMI, and IRQ6 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).
3. In standby mode, $t_{MRESW} = t_{OSC2}$ (10 ms).
4. Input t_{RESW1} which satisfies all the conditions.

Table 29.11 Control Signal Timing (SH7239B and SH7237B)

		Bϕ = 50 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{\text{RES}}$ pulse width (except during flash memory programming/erasing)	t_{RESW1}	20*[2]*[4]	—	t_{cyc}	Figures 29.3 to 29.6
		1.5*[4]	—	µs	
$\overline{\text{RES}}$ pulse width (during flash memory programming/erasing)	t_{RESW2}	100	—	µs	
$\overline{\text{RES}}$ setup time*[1]	t_{RESS}	100	—	ns	
$\overline{\text{RES}}$ hold time	t_{RESH}	15	—	ns	
$\overline{\text{MRES}}$ pulse width	t_{MRESW}	20*[3]	—	t_{cyc}	
$\overline{\text{MRES}}$ setup time	t_{MRESS}	120	—	ns	
$\overline{\text{MRES}}$ hold time	t_{MRESH}	15	—	ns	
MD0, FWE setup time	t_{MDS}	20	—	t_{cyc}	Figure 29.6
NMI setup time*[1]	t_{NMIS}	100	—	ns	Figure 29.7
NMI hold time	t_{NMIH}	10	—	ns	
IRQ6 to IRQ0 setup time*[1]	t_{IRQS}	35	—	ns	
IRQ6 to IRQ0 hold time	t_{IRQH}	10	—	ns	
IRQ pulse width	t_{IRQW}	4	—	t_{cyc}	
NMI pulse width	t_{NMIW}	4	—	t_{cyc}	
$\overline{\text{IRQOUT}}$ output delay time	t_{IRQOD}	—	100	ns	Figure 29.9

Notes: 1. $\overline{\text{RES}}$, NMI, and IRQ6 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).
3. In standby mode, $t_{MRESW} = t_{OSC2}$ (10 ms).
4. Input t_{RESW1} which satisfies all the conditions.

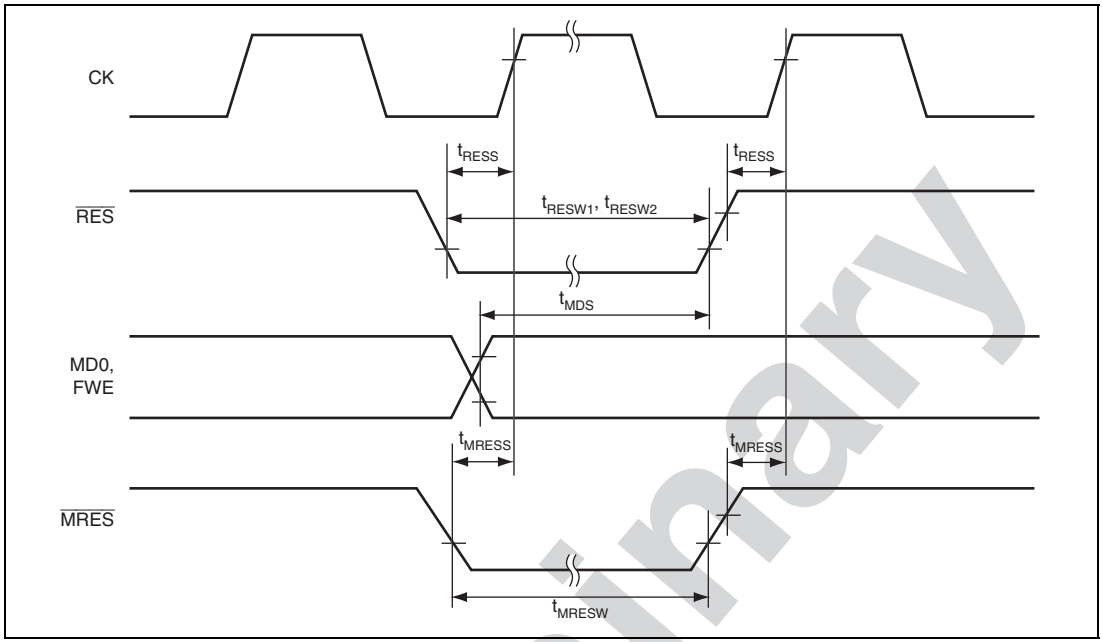


Figure 29.6 Reset Input Timing

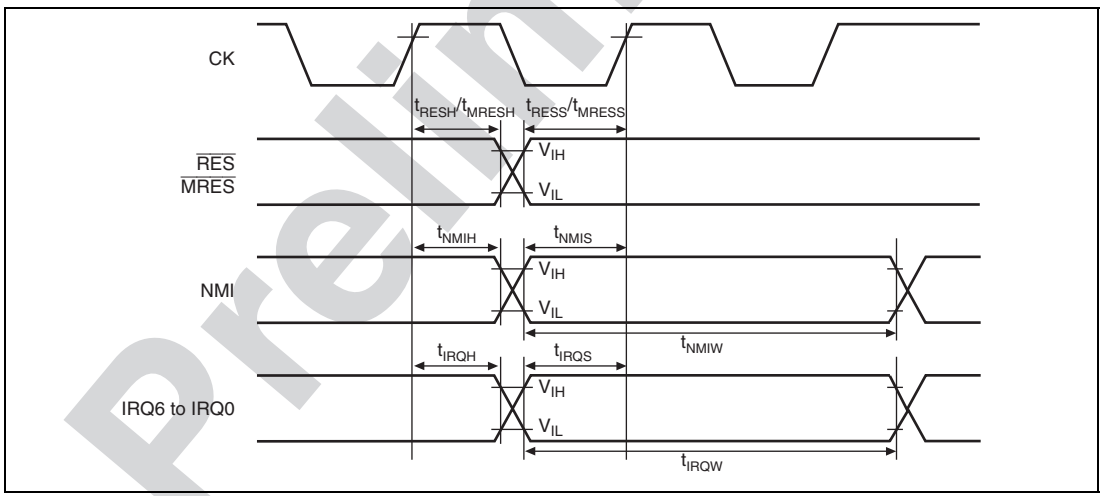


Figure 29.7 Interrupt Signal Input Timing

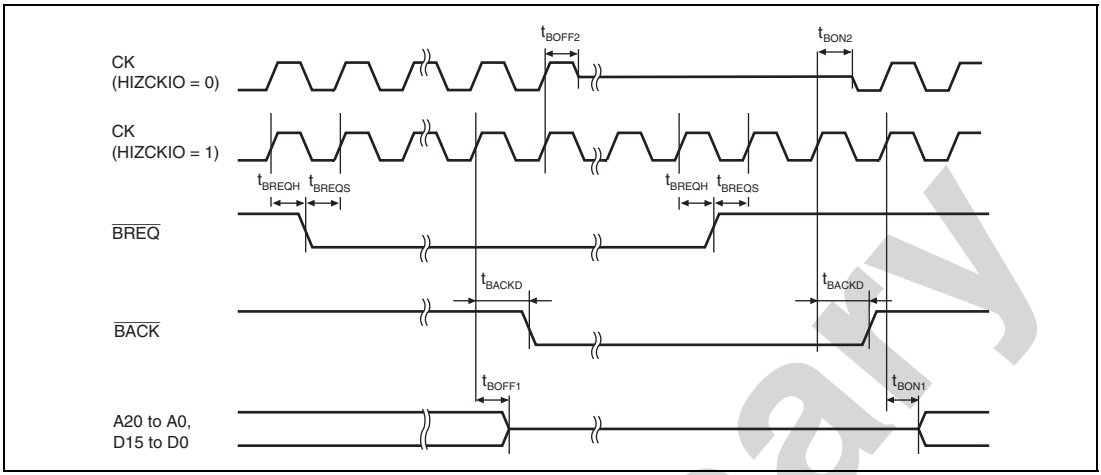


Figure 29.8 Bus Release Timing

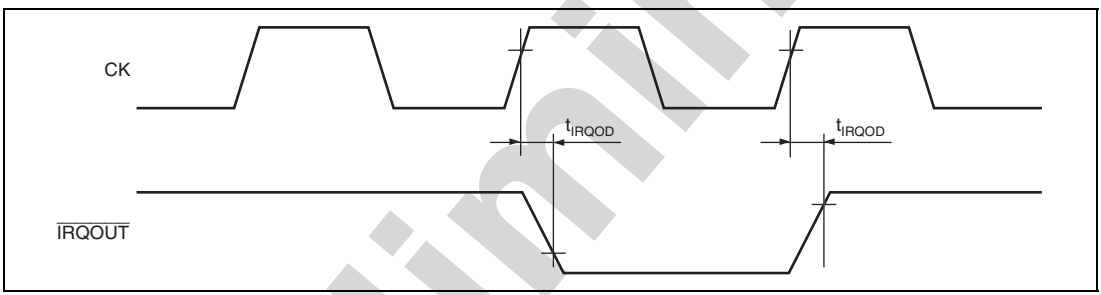


Figure 29.9 Interrupt Signal Output Timing

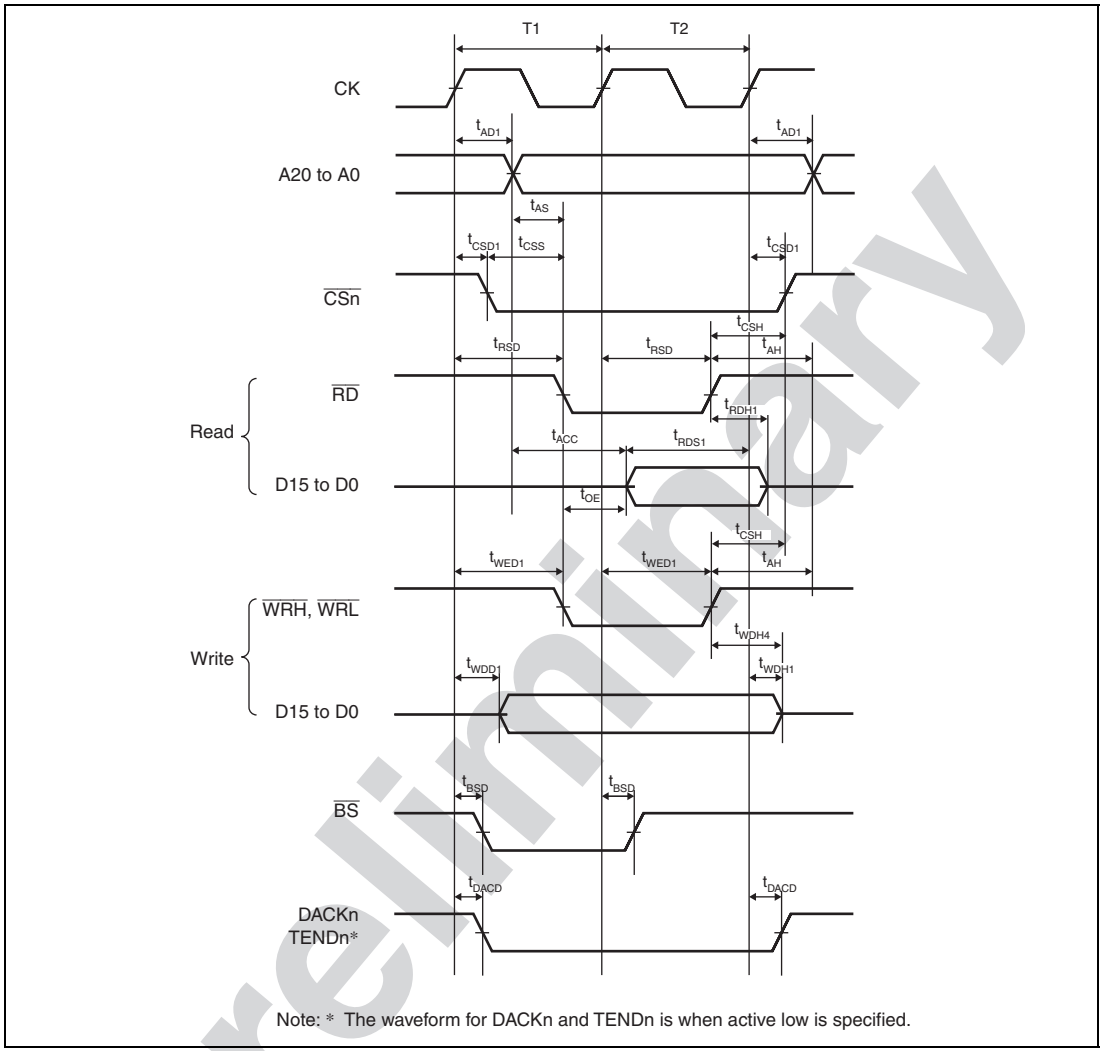
29.3.3 Bus Timing (SH7239A and SH7237A only)

Table 29.12 Bus Timing

		Bϕ = 40 MHz*1			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	18	ns	Figures 29.10 to 29.14
Address setup time	t_{AS}	0	—	ns	Figures 29.10 to 29.13
Address hold time	t_{AH}	0	—	ns	Figures 29.10 to 29.13
$\overline{BS}$ delay time	t_{BSD}	—	18	ns	Figures 29.10 to 29.14
$\overline{CS}$ delay time 1	t_{CSD1}	1	18	ns	Figures 29.10 to 29.14
$\overline{CS}$ setup time	t_{CSS}	0	—	ns	Figures 29.10 to 29.13
$\overline{CS}$ hold time	t_{CSH}	0	—	ns	Figures 29.10 to 29.13
Read strobe delay time	t_{RSD}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 29.10 to 29.14
Read data setup time 1	t_{RDS1}	$1/2t_{cyc} + 18$	—	ns	Figures 29.10 to 29.14
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 29.10 to 29.14
Write enable delay time 1	t_{WED1}	$1/2t_{cyc} + 1$	$1/2t_{cyc} + 18$	ns	Figures 29.10 to 29.14
Write data delay time 1	t_{WDD1}	—	18	ns	Figures 29.10 to 29.14
Write data hold time 1	t_{WDH1}	1	18	ns	Figures 29.10 to 29.14
Write data hold time 4	t_{WDH4}	0	18	ns	Figures 29.10 to 29.14
Read data access time	t_{ACC}*3	$t_{cyc}(n + 1.5) - 32$*2	—	ns	Figures 29.10 to 29.13
Access time from read strobe	t_{OE}*3	$t_{cyc}(n + 1) - 32$*2	—	ns	Figures 29.10 to 29.13

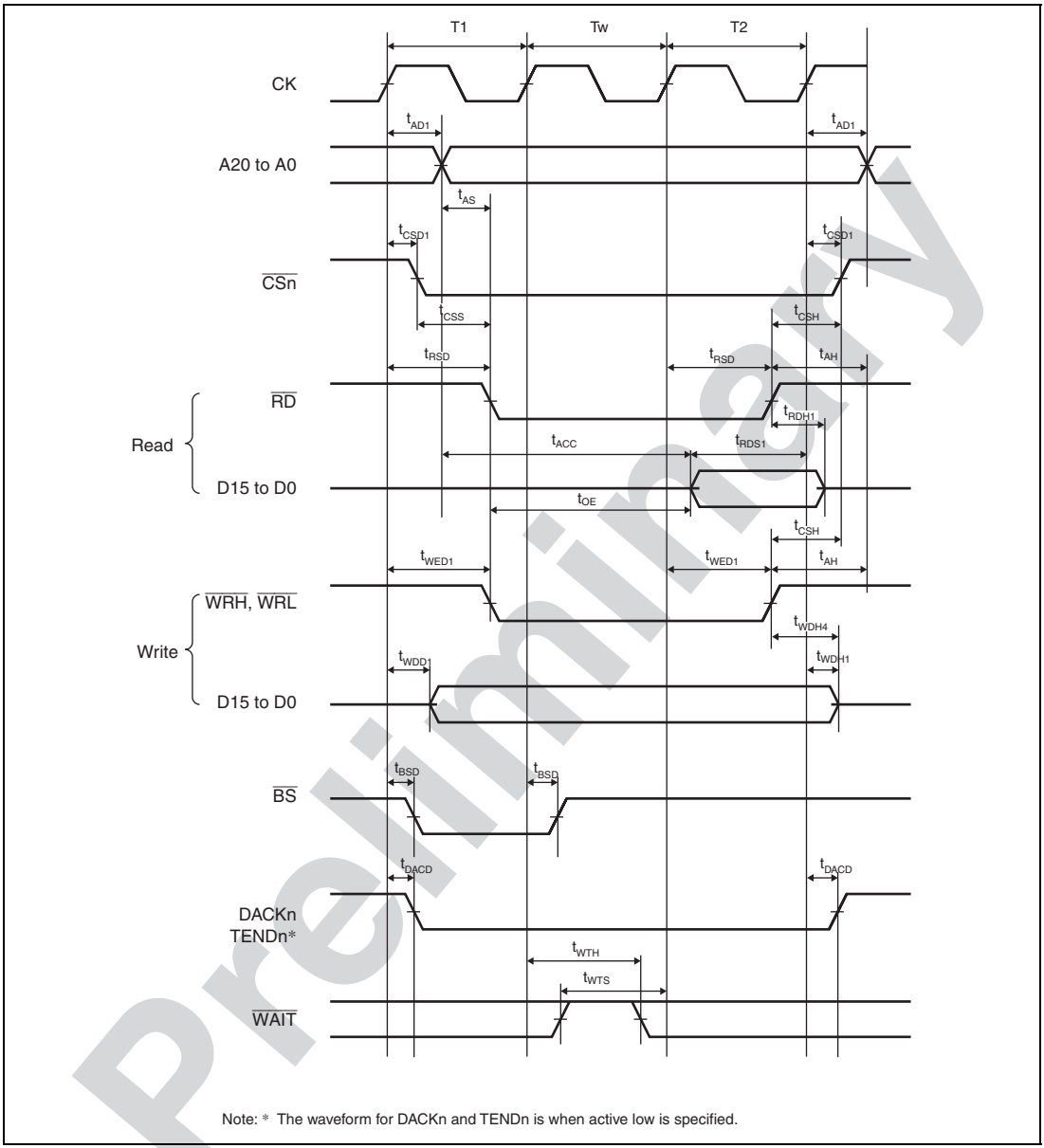
Item	Symbol	Bϕ = 40 MHz*1 Min.	Max.	Unit	Figure
$\overline{\text{WAIT}}$ setup time	t_{WTS}	$1/2t_{cyc}$ + 15	—	ns	Figures 29.11 to 29.14
$\overline{\text{WAIT}}$ hold time	t_{WTH}	$1/2t_{cyc}$ + 2	—	ns	Figures 29.11 to 29.14
$\overline{\text{AH}}$ delay time	t_{AHD}	$1/2t_{cyc}$ + 1	$1/2t_{cyc}$ + 18	ns	Figure 29.14
Multiplexed address delay time	t_{MAD}	–	18	ns	Figure 29.14
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 29.14
DACK, TEND delay time	t_{DACD}	—	Refer to peripheral modules	ns	Figures 29.10 to 29.14

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 indicates the number of wait cycles.
3. When the access time is satisfied, t_{RDS1} need not be satisfied.



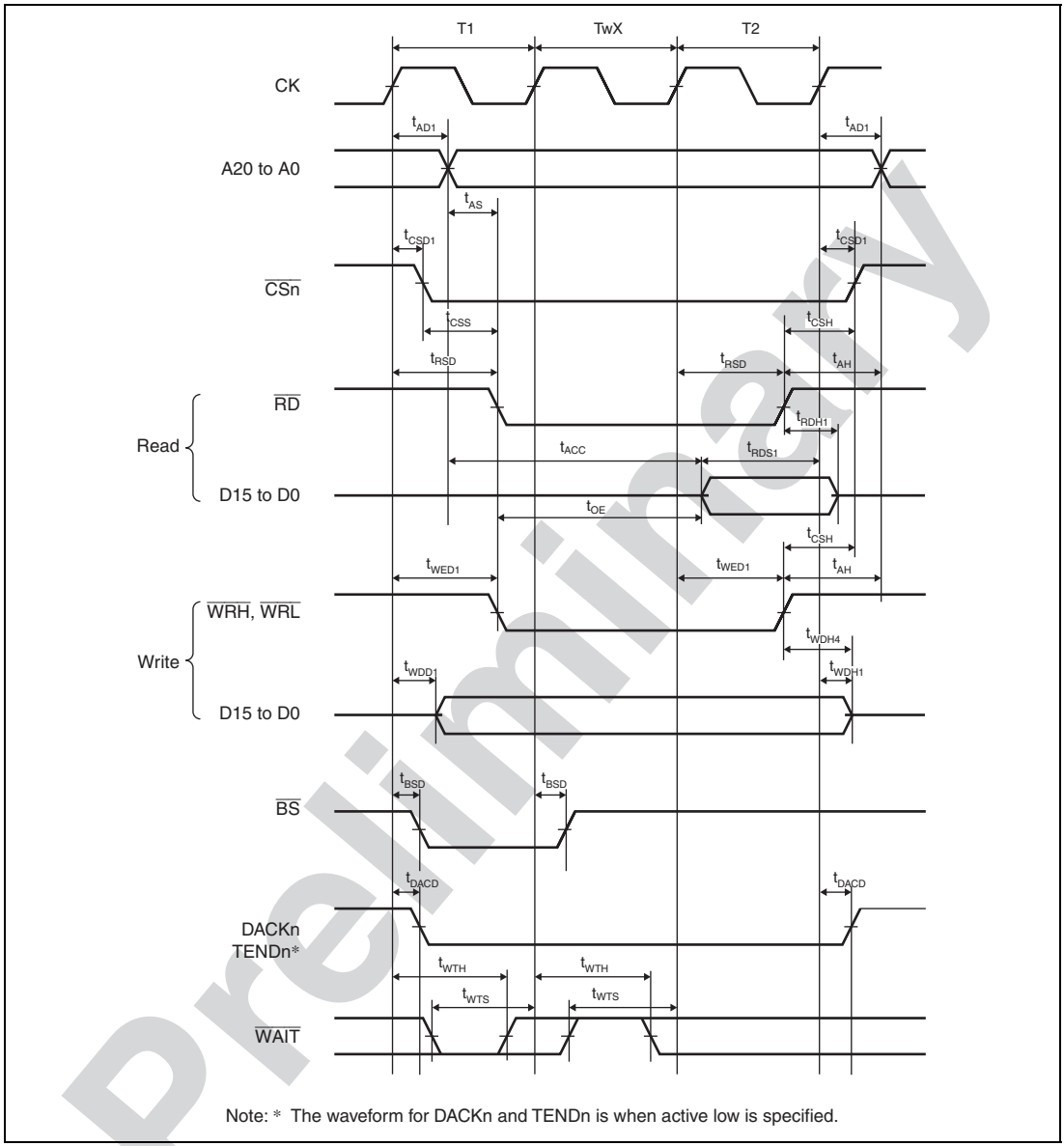
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 29.10 Basic Bus Timing for Normal Space (No Wait)



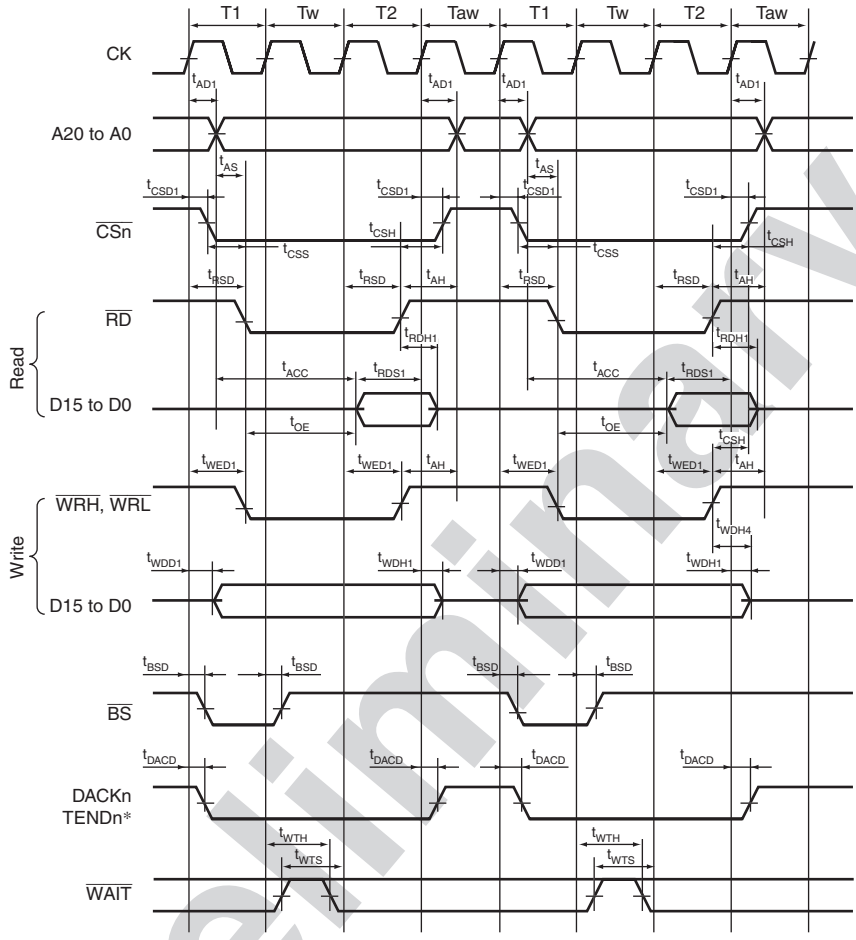
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 29.11 Basic Bus Timing for Normal Space (One Software Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 29.12 Basic Bus Timing for Normal Space (One External Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 29.13 Basic Bus Timing for Normal Space (One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)

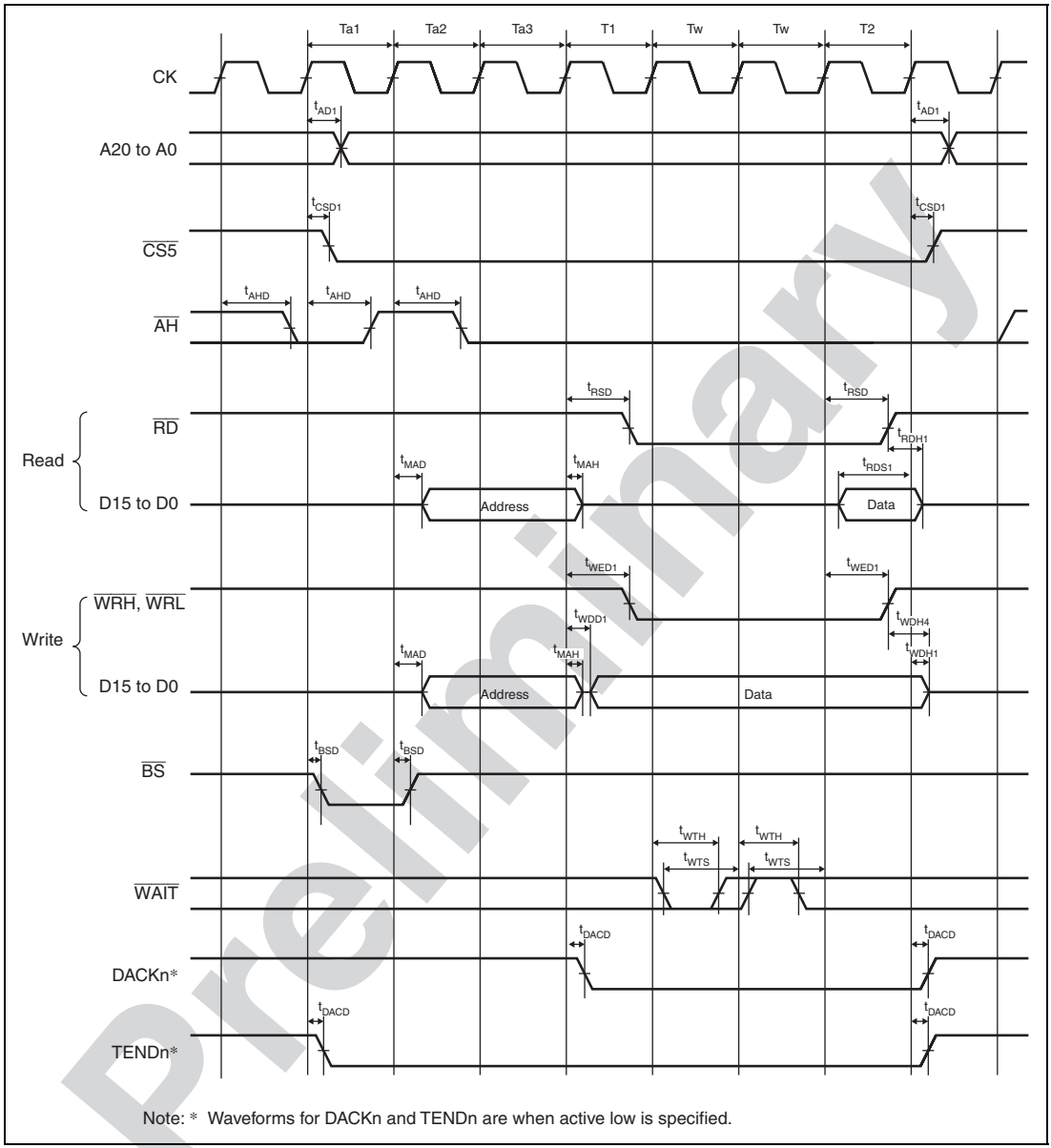


Figure 29.14 MPX-I/O Interface Bus Cycle
(Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)

29.3.4 UBC Trigger Timing

Table 29.13 UBC Trigger Timing

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	20	ns	Figure 29.15

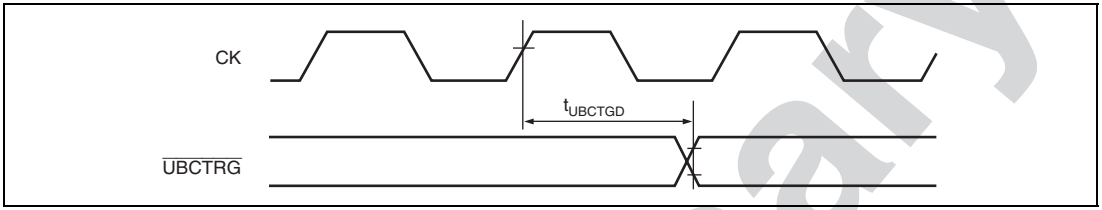
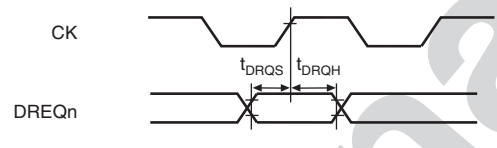


Figure 29.15 UBC Trigger Timing

29.3.5 DMAC Module Timing

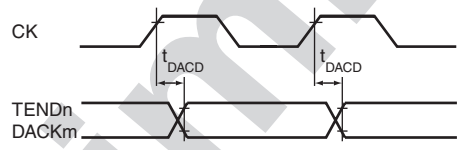
Table 29.14 DMAC Module Timing

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	20	—	ns	Figure 29.16
DREQ hold time	t_{DRQH}	20	—		
DACK, TEND delay time	t_{DACD}	—	20		Figure 29.17



Note: n = 0 to 3

Figure 29.16 DREQ Input Timing



Note: n = 0, 1
m = 0 to 3

Figure 29.17 DACK, TEND Output Timing

29.3.6 MTU2, MTU2S Module Timing

Table 29.15 MTU2, MTU2S Module Timing

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	Figure 29.18
Input capture input setup time	t_{TICS}	20	—	ns	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 29.19
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 (Mϕ) cycle.

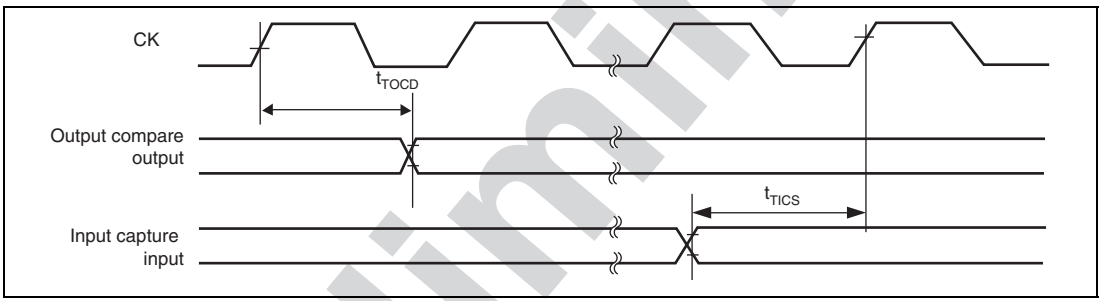


Figure 29.18 MTU2, MTU2S Input/Output Timing

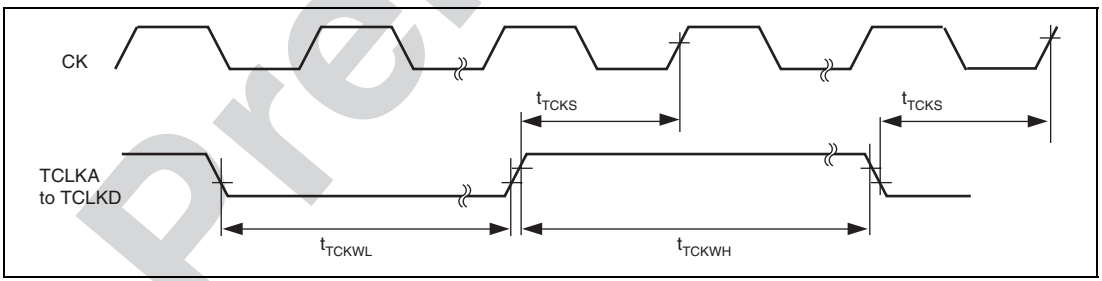


Figure 29.19 MTU2, MTU2S Clock Input Timing

29.3.7 POE2 Module Timing

Table 29.16 POE2 Module Timing

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{POE}}$ input setup time	t_{POES}	50	—	ns	Figure 29.20
$\overline{\text{POE}}$ input pulse width	t_{POEW}	1.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

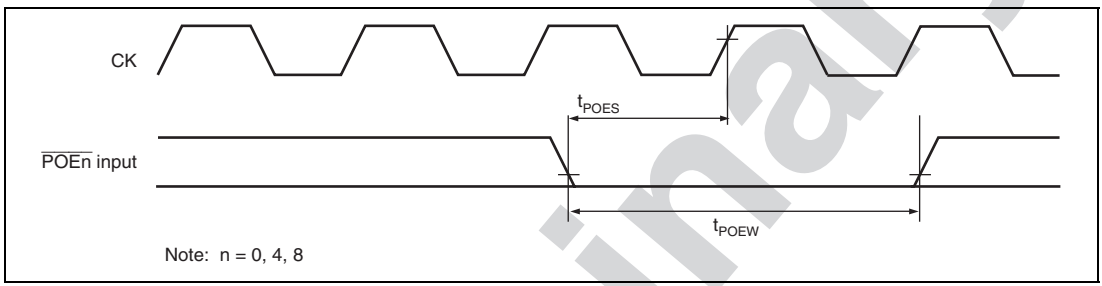


Figure 29.20 POE2 Input Timing

29.3.8 Watchdog Timer Timing

Table 29.17 Watchdog Timer Timing

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	50	ns	Figure 29.21

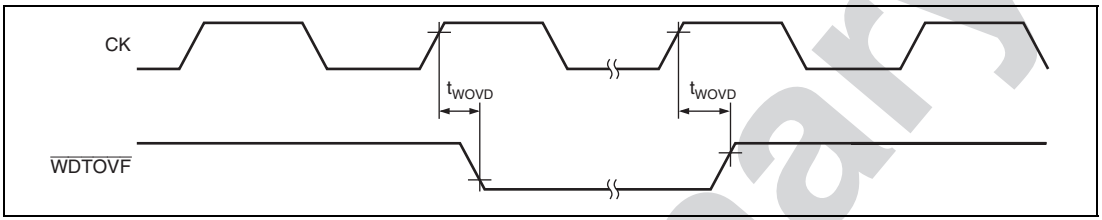


Figure 29.21 Watchdog Timer Timing

29.3.9 SCI Module Timing

Table 29.18 SCI Module Timing

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle (asynchronous)		t_{Scyc}	4	—	t_{pcyc}	Figure 29.22
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 29.23
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.

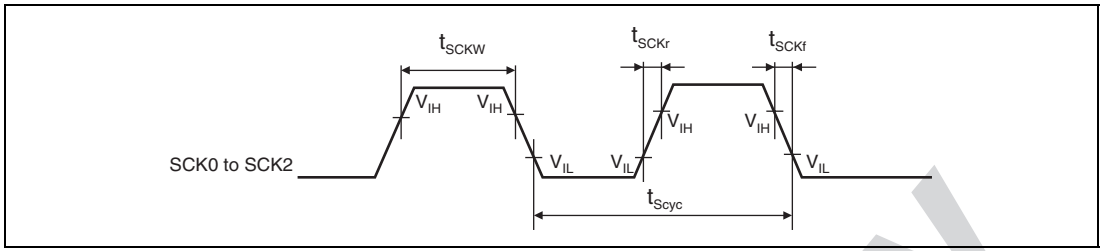


Figure 29.22 Input Clock Timing

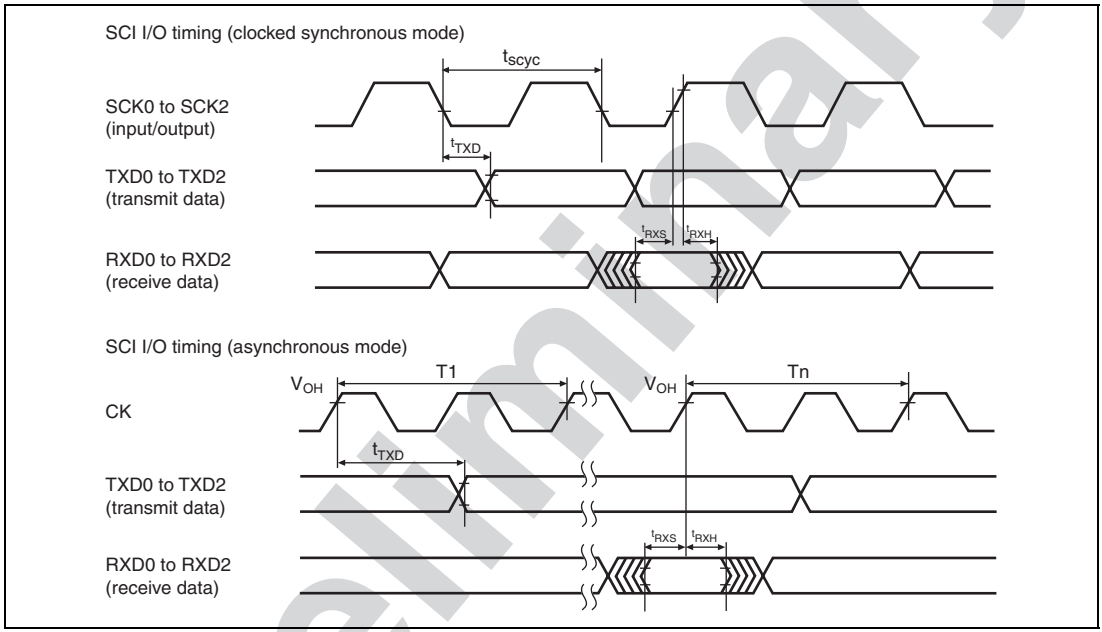


Figure 29.23 SCI Input/Output Timing

29.3.10 SCIF Module Timing

Table 29.19 SCIF Module Timing

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	6	—	t_{pcyc}	Figure 29.24
	(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 29.25
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.

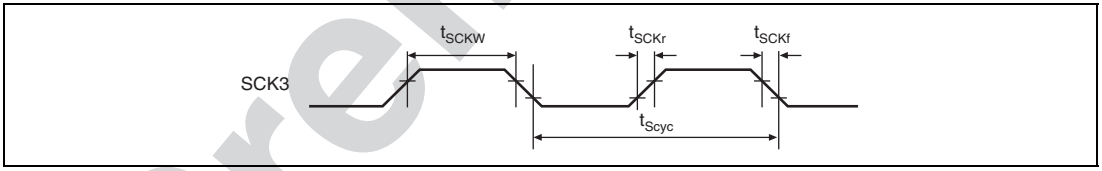


Figure 29.24 Input Clock Timing

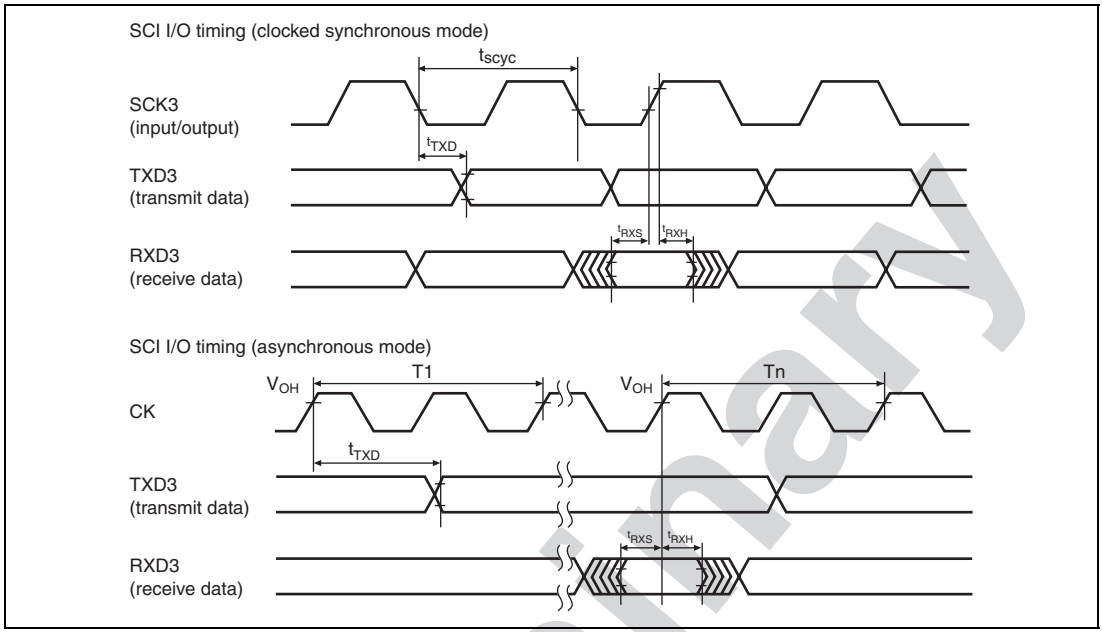


Figure 29.25 SCIF Input/Output Timing

29.3.11 RSPI Timing

Table 29.20 RSPI Timing

Item		Symbol	Min.	Typ.	Max.	Unit	Figure
RSPCK clock cycle*1	Master	t_{SPcyc}	2	—	4096	t_{Pcyc}	Figure 29.26
	Slave		8	—	4096		
RSPCK clock high pulse width	Master	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) /2 - 5$	—	—	ns	
	Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) /2$	—	—		
RSPCK clock low pulse width	Master	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) /2 - 5$	—	—	ns	
	Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) /2$	—	—		
PSPCK clock rise/ fall time*2	Output	t_{SPCKR}, t_{SPCKF}	—	—	7	ns	
	input		—	—	1	t_{Pcyc}	
Data input setup time	Master	t_{SU}	23	—	—	ns	Figures 29.27 to 29.30
	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}	—	—	8	ns	
	Slave		—	—	$3t_{Pcyc} + 18$		
Data output hold time	Master	t_{OH}	0	—	—	ns	
	Slave		0	—	—		

Item		Symbol	Min.	Typ.	Max.	Unit	Figure
Continuous transmission delay time	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	—	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	Figures 29.27 to 29.30
	Slave		$4 \times t_{pcyc}$	—	—		
MOSI, MISO rise/ fall time*2	Master	t_{DR}, t_{DF}	—	—	7	ns	
	Slave		—	—	1	t_{Pcyc}	
SSL rise/fall time	Master	t_{SSLR}, t_{SSLF}	—	—	7	ns	
	Slave		—	—	1	t_{Pcyc}	
Slave access time		t_{SA}	—	—	4	t_{Pcyc}	Figures 29.29 and 29.30
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.

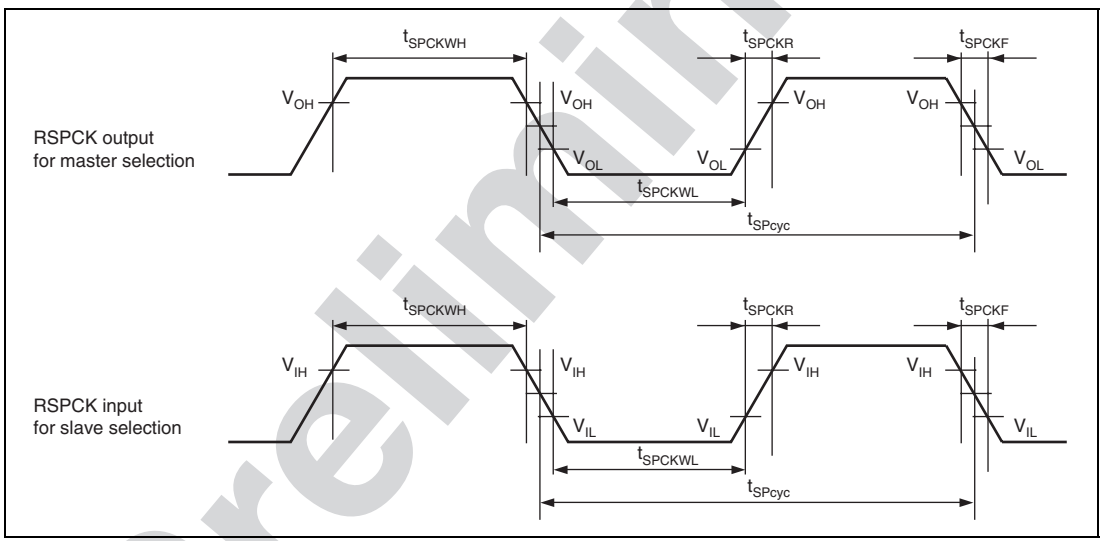


Figure 29.26 SPI Clock Timing

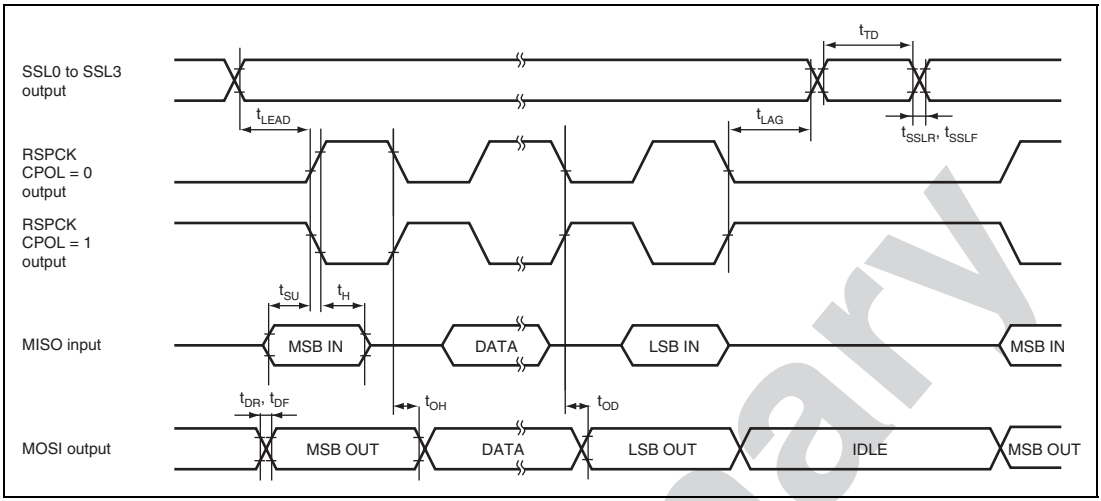


Figure 29.27 SPI Timing (Master, CPHA = 0)

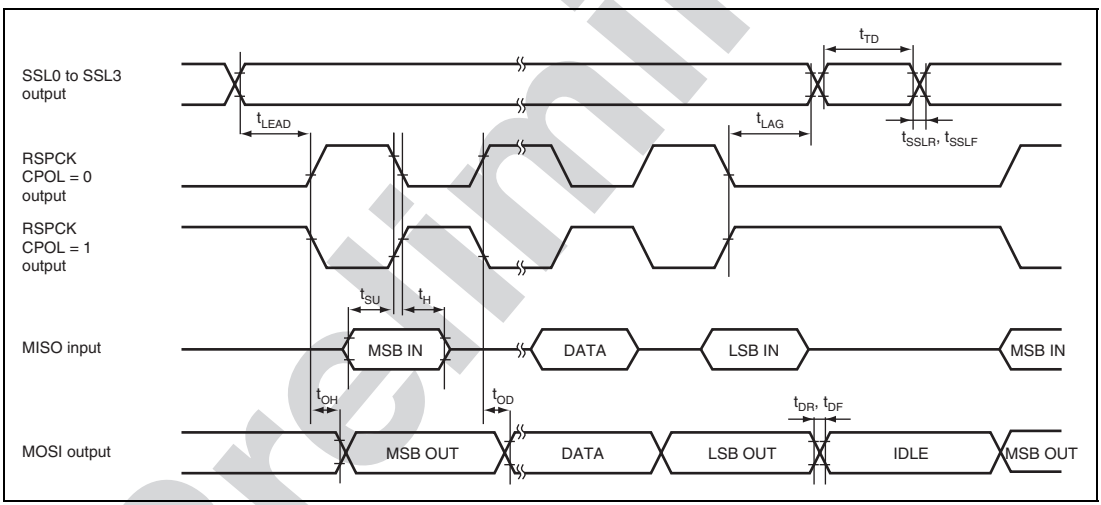


Figure 29.28 SPI Timing (Master, CPHA = 1)

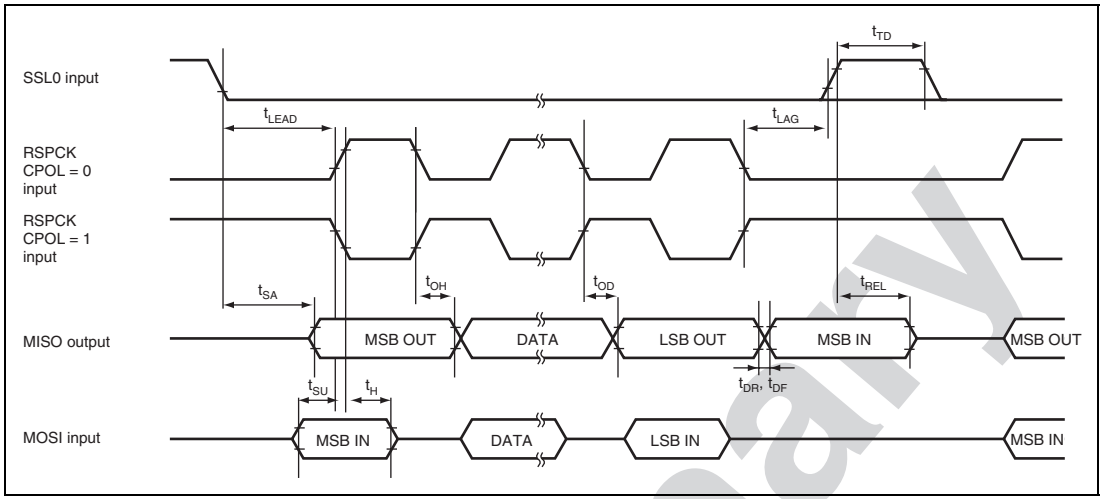


Figure 29.29 SPI Timing (Slave, CPHA = 0)

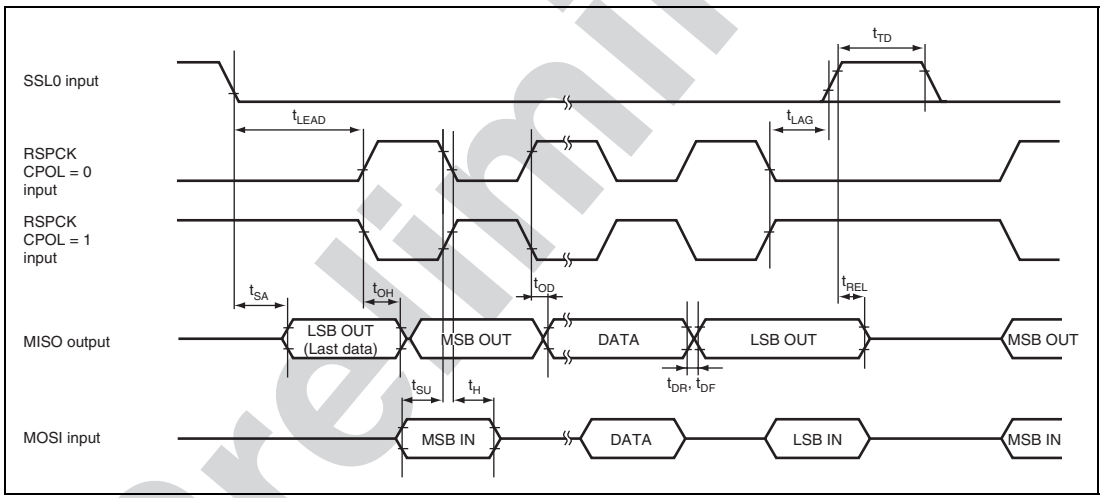


Figure 29.30 SPI Timing (Slave, CPHA = 1)

29.3.12 Controller Area Network (RCAN-ET) Timing

Table 29.21 Controller Area Network (RCAN-ET) Timing

Item	Symbol	Min.	Max.	Unit	Figure
Transmit data delay time	t_{CTxD}	—	100	ns	Figure 29.31
Receive data setup time	t_{CRxS}	100	—	ns	
Receive data hold time	t_{CRxH}	100	—	ns	

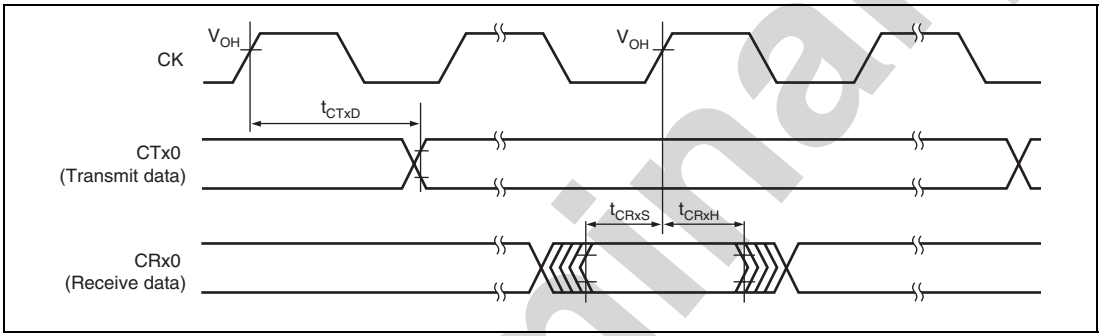


Figure 29.31 RCAN-ET Input/Output Timing

29.3.13 A/D Trigger Input Timing

Table 29.22 A/D Trigger Input Timing

Module	Item		Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	P:A clock ratio = 1:1	t_{TRGS}	20	—	ns	Figure 29.32
		P:A clock ratio = 2:1		t_{pcyc} + 20	—		
		P:A clock ratio = 4:1		$3 \times t_{pcyc}$ + 20	—		

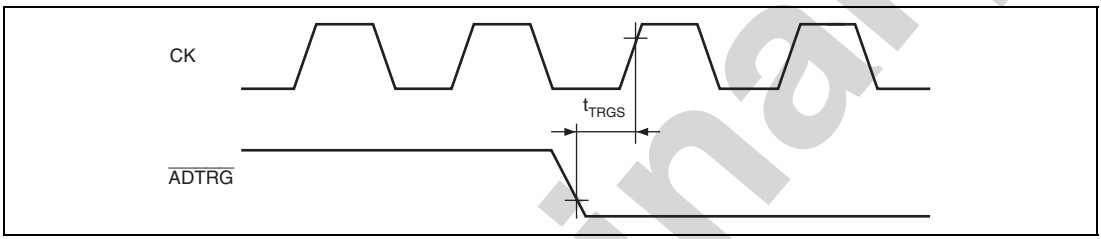


Figure 29.32 A/D Converter External Trigger Input Timing

29.3.14 I/O Port Timing

Table 29.23 I/O Port Timing

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	50	ns	Figure 29.33
Input data setup time	t_{PORTS}	20	—		
Input data hold time	t_{PORTH}	20	—		

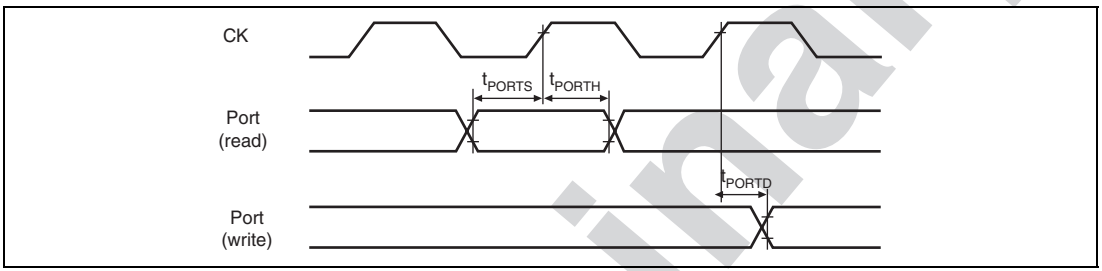


Figure 29.33 I/O Port Timing

29.3.15 H-UDI Related Pin Timing

Table 29.24 H-UDI Related Pin Timing

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	50*	—	ns	Figure 29.34
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 29.35
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	ns	

Note: * This value must exceed the cycle time for the peripheral clock (Pϕ).

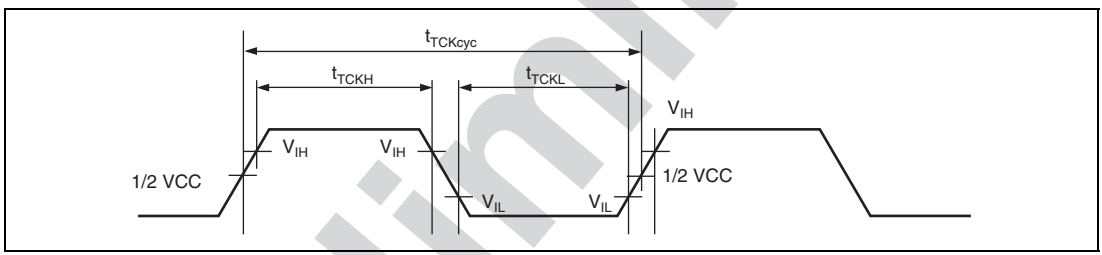


Figure 29.34 TCK Input Timing

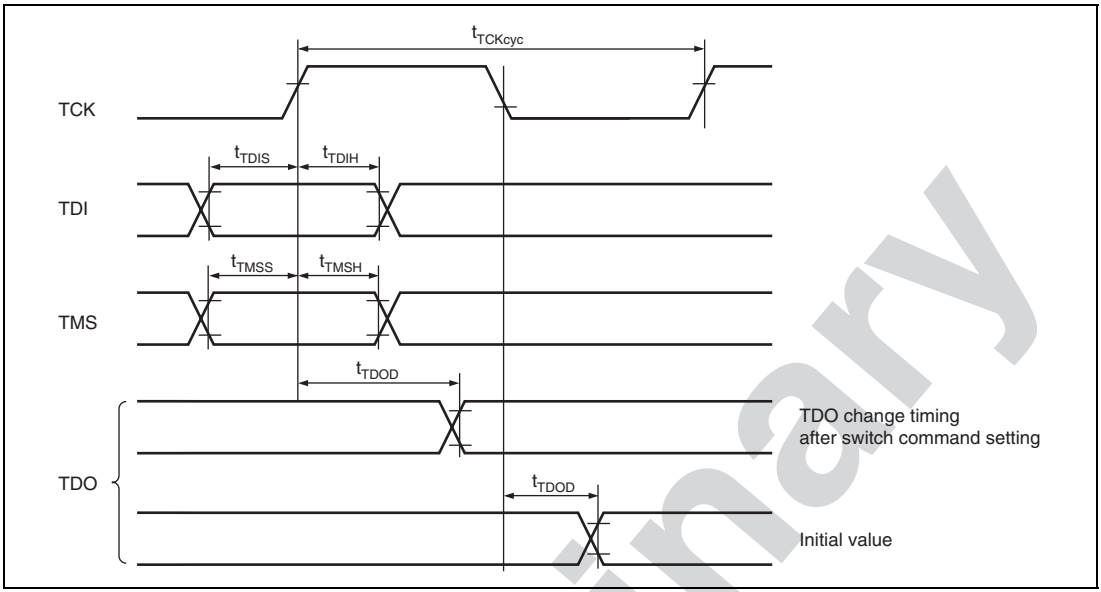
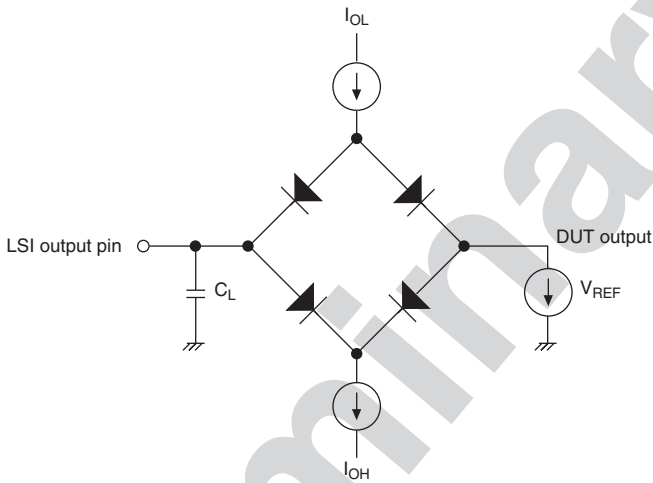


Figure 29.35 H-UDI Data Transmission Timing

29.3.16 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:
20 pF: CK
30 pF: All pins
2. Test conditions include I_{OL} = 1.6 mA and I_{OH} = −200 µA.

Figure 29.36 Output Load Circuit

29.4 A/D Converter Characteristics

Table 29.25 A/D Converter Characteristics

Item		Min.	Typ.	Max.	Unit	Test condition
Resolution		—	12.0	—	bits	
Conversion time (operating at 40-MHz AD clock)		1.25	—	—	µs	Sample & hold circuits not in use
		2	—	—	µs	Sample & hold circuits in use
Conversion time (operating at 50-MHz AD clock)		1.0	—	—	µs	Sample & hold circuits not in use
		1.6	—	—	µs	Sample & hold circuits 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

29.5 Flash Memory Characteristics

Table 29.26 ROM (Flash Memory for Code Storage) Characteristics

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time	256 bytes	t_{P256}	—	2	12	ms	Pϕ = 50 MHz, 40 MHz $N_{PEC} \leq 100$
	8 Kbytes	t_{P8K}	—	45	100	ms	
	256 bytes	t_{P256}	—	2.4	14.4	ms	Pϕ = 50 MHz, 40 MHZ $N_{PEC} > 100$
	8 Kbytes	t_{P8K}	—	54	120	ms	
Erase time	8 Kbytes	t_{E8K}	—	50	120	ms	Pϕ = 50 MHz, 40 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, 40 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}	—	—	120	µs	Figure 29.37
First suspend delay time during erasing (in suspension priority mode)		t_{SESD1}	—	—	220 (Pϕ = 20 MHz), 130 (Pϕ = 40 MHz), 120 (Pϕ = 50 MHz)	µs	Pϕ = 50 MHz, 40 MHz
Second suspend delay time during erasing (in suspension priority mode)		t_{SESD2}	—	—	1.7	ms	
Suspend delay time during erasing (in erasure priority mode)		t_{SEED}	—	—	1.7	ms	
Data hold time*3		t_{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.

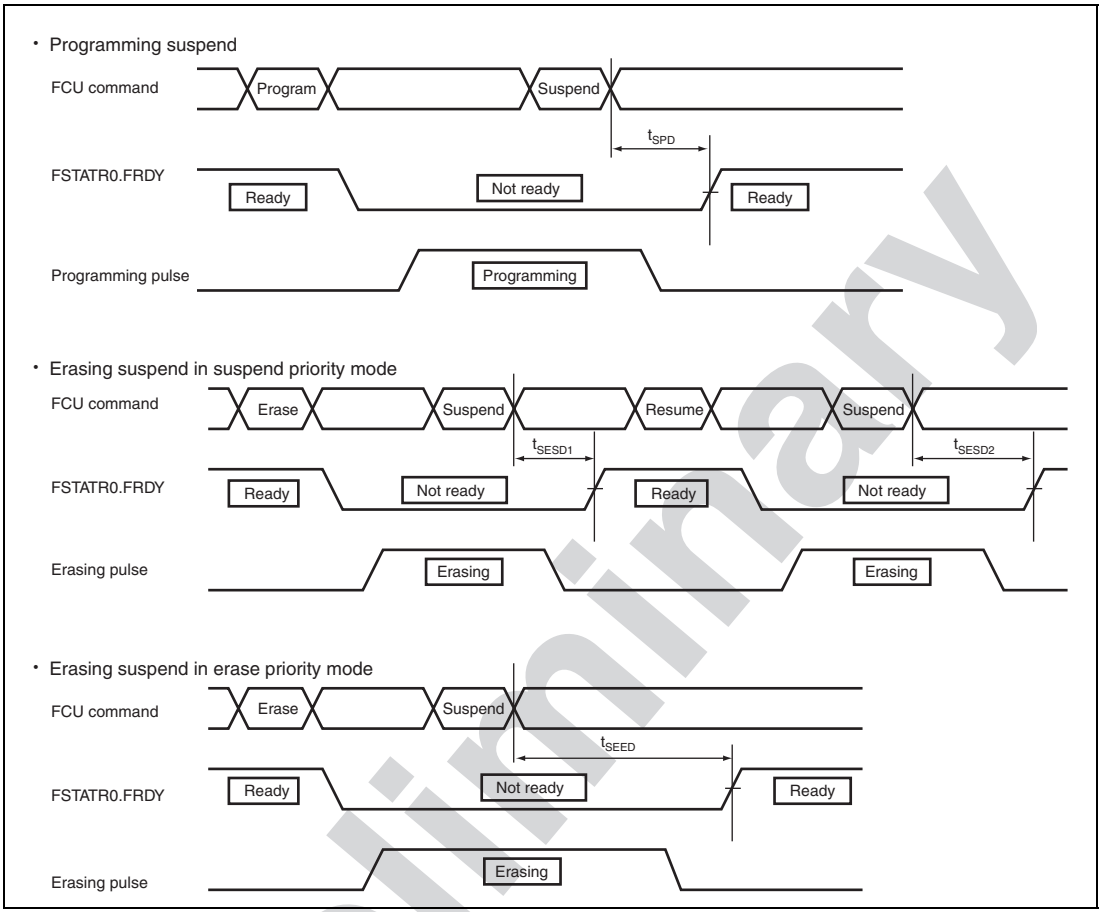


Figure 29.37 Flash Programming/Erasing Suspend Timing

29.6 FLD Characteristics

Table 29.27 FLD (Flash Memory for Data Storage) Characteristics

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time	8 bytes	t_{P8}	—	0.4	2	ms	Pϕ = 50 MHz, 40 MHz
	128 bytes	t_{P128}	—	1	5	ms	
Erasure time	2 Kbytes	t_{E8K}	—	70	250	ms	Pϕ = 50 MHz, 40 MHz
Blank check time	8 bytes	t_{BC8}	—	—	30	µs	Pϕ = 50 MHz, 40 MHz
	2 Kbytes	t_{BC8K}	—	—	0.7	ms	
Rewrite/erase cycle*1		N_{PEC}	30000*2	—	—	Times	
Suspend delay time during writing		t_{SPD}	—	—	120	µs	Figure 29.37
First suspend delay time during erasing (in suspension priority mode)		t_{SESD1}	—	—	220 (Pϕ = 20 MHz), 130 (Pϕ = 40 MHz), 120 (Pϕ = 50 MHz)	µs	Pϕ = 50 MHz, 40 MHz
Second suspend delay time during erasing (in suspension priority mode)		t_{SESD2}	—	—	1.7	ms	
Suspend delay time during erasing in erasure priority mode		t_{SEED}	—	—	1.7	ms	
Data hold time*3		t_{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 16 times for different addresses in 2-Kbyte block and then the entire block is erased, the rewrite/erase cycle is counted as one. However, writing to the same address for several times as one erasing is not enabled (over writing is prohibited).

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.

29.7 Usage Note

29.7.1 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 29.38. 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 4.8, Notes on Board Design.

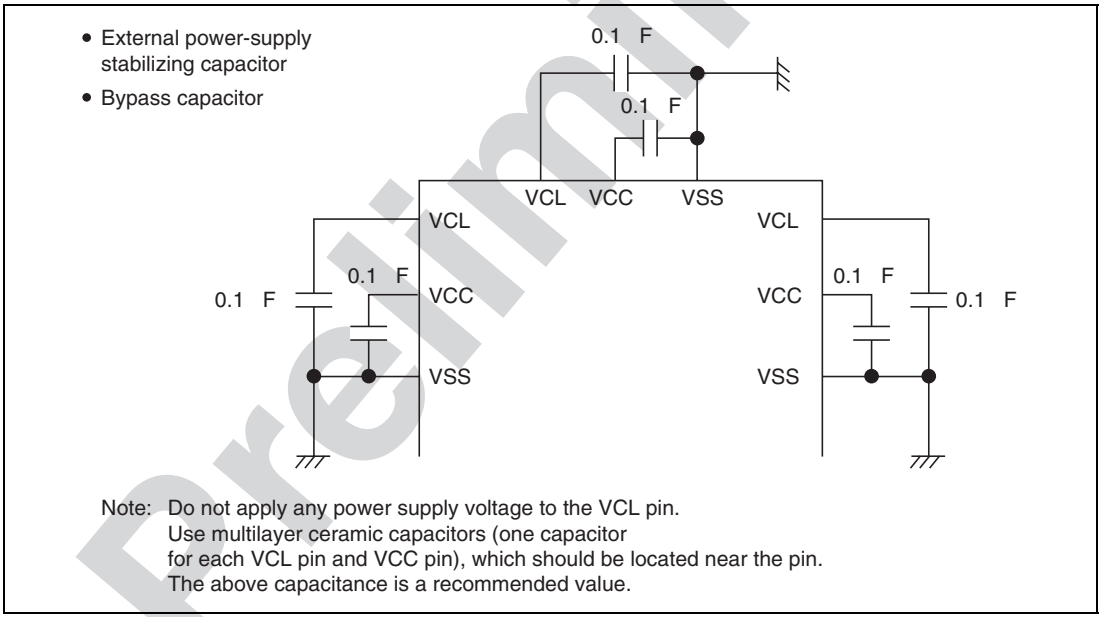


Figure 29.38 Connection of Capacitors

Appendix

A. Pin States

Pin initial states differ according to MCU operating modes. Refer to section 21, Pin Function Controller (PFC), for details.

Table A.1 Pin States

Pin Function		Pin State							
		Reset State			Power-Down State				
		Power-On							
Type	Pin Name	MCU Extension Mode 2*10	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release*10	Oscillation Stop Detected	POE Function Used
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	$\overline{\text{RES}}$	I		I	I	I	I	I	I
	$\overline{\text{MRES}}$	Z		I	I*6	I	I	I*6	I
	$\overline{\text{WDTOVF}}$	O*8		O	O	O	O	O	O
	$\overline{\text{BREQ}}$	Z		I	Z	I	I	I	I
	$\overline{\text{BACK}}$	Z		O	Z	O	L	O	O
Operating mode control	MD0	I		I	I	I	I	I	I
	$\overline{\text{ASEMD0}}$	I*9		I*9	I*9	I*9	I*9	I*9	I*9
	FWE	I		I	I	I	I	I	I
Interrupt	NMI	I		I	I	I	I	I	I
	IRQ0 to IRQ6	Z		I	I	I	I	I	I
	$\overline{\text{IRQOUT}}$ (PE15)	Z		O	Z (MZIZEH in HCPCR = 0)	O	O	O*6	O
					H*1 (MZIZEH in HCPCR = 1)				
	$\overline{\text{IRQOUT}}$ (PE1)	Z		O	H*1	O	O	O	O

Pin Function		Pin State							
		Reset State			Power-Down State				
		Power-On							
Type	Pin Name	MCU Extension Mode 2※10	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release※10	Oscillation Stop Detected	POE Function Used
Address bus	A0 to A20	Z		O	Z※3	O	Z	O	O
Data bus	D0 to D9	Z		I/O	Z	I/O	Z	I/O	I/O
	D10 to D15	Z		I/O	Z	I/O	Z	I/O※5	I/O
Bus control	$\overline{\text{WAIT}}$	Z		I	Z	I	Z	I	I
	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	Z		O	Z※3	O	Z	O	O
	$\overline{\text{CS3}}$ to $\overline{\text{CS6}}$	Z		O	Z※3	O	Z	O	O
	$\overline{\text{BS}}$	Z		O	Z※3	O	Z	O	O
	$\overline{\text{AH}}$	Z		O	Z※3	O	Z	O	O
	$\overline{\text{RD}}$	Z		O	Z※3	O	Z	O	O
	$\overline{\text{WRH}}$, $\overline{\text{WRL}}$	Z		O	Z※3	O	Z	O	O
DMAC	DREQ0 (PE0), DREQ1 (PE2)	Z		I	Z	I	I	I※7	I
	DACK0 (PE14), DACK1 (PE15), DACK2, DACK3	Z		O	Z (MZIZEH in HCPCR = 0)	O	O	O※6	O
					O※1 (MZIZEH in HCPCR = 1)				
	TEND0 (PE1), TEND1 (PE3)	Z		O	Z (MZIZEL in HCPCR = 0)	O	O	O※7	O
					O※1 (MZIZEL in HCPCR = 1)				

Pin Function		Pin State							
		Reset State			Power-Down State				
		Power-On							
Type	Pin Name	MCU Extension Mode 2※10	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release※10	Oscillation Stop Detected	POE Function Used
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)	I/O	I/O	I/O※7	Z
					K※1 (MZIZEL in HCPCR = 1)				
	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
	TIOC1B (PE5), TIOC2A (PE6)	Z		I/O	Z (MZIZEL in HCPCR = 0)	I/O	I/O	I/O※7	I/O
					K※1 (MZIZEL in HCPCR = 1)				
	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)	I/O	I/O	I/O※6	Z
					K※1 (MZIZEH in HCPCR = 1)				

Pin Function		Pin State							
		Reset State			Power-Down State				
		Power-On							
Type	Pin Name	MCU Extension Mode 2*10	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release*10	Oscillation Stop Detected	POE Function Used
MTU2	TIOC4A, TIOC4B, TIOC4C, TIOC4D	Z		I/O	Z (MZIZEH in HCPCR = 0)	I/O	I/O	I/O*6	Z
					K*1 (MZIZEH in HCPCR = 1)				
	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)	I/O	I/O	I/O*5	Z
					K*1 (MZIZDL 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*7	Z
					K*1 (MZIZEL in HCPCR = 1)				
	TIC5US, TIC5VS, TIC5WS	Z		I	Z	I	I	I	I
POE2	$\overline{\text{POE0}}$, $\overline{\text{POE4}}$, $\overline{\text{POE8}}$	Z		I	Z	I	I	I	I
SCI	SCK0 to SCK2	Z		I/O	K*1	I/O	I/O	I/O	I/O
	RXD0 to RXD2	Z		I	Z	I	I	I	I
	TXD0 to TXD2	Z		O	O*1	O	O	O	O

Pin Function		Pin State							
		Reset State			Power-Down State				
		Power-On							
Type	Pin Name	MCU Extension Mode 2※10	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release※10	Oscillation Stop Detected	POE Function Used
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※7	I
	TXD3 (PE5)	Z		O	Z (MZIZEL in HCPCR = 0)	O	O	O※7	O
					O※1 (MZIZEL in HCPCR = 1)				
	TXD3 (PB3)	Z		O	O※1	O	O	O	O
RSPI	RSPCK	Z		I/O	K※1	I/O	I/O	I/O	I/O
RSPI	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
UBC	$\overline{\text{UBCTRG}}$	Z		O	O※1	O	O	O	O
A/D converter	AN0 to AN15	Z		I	Z	I	I	I	I
	$\overline{\text{ADTRG}}$	Z		I	Z	I	I	I	I
RCAN-ET	CRx0	Z		I	Z	I	I	I	I
	CTx0	Z		O	O※1	O	O	O	O
I/O port	PA0, PA1, PA6 to PA9, PA15 to PA18	Z		I/O	K※1	I/O	I/O	I/O	I/O
	PB0 to PB4, PB16 to PB21	Z		I/O	K※1	I/O	I/O	I/O	I/O
	PC0 to PC15	Z		I/O	K※1	I/O	I/O	I/O	I/O
	PD0 to PD9	Z		I/O	K※1	I/O	I/O	I/O	I/O

Pin Function		Pin State							
		Reset State			Power-Down State				
		Power-On							
Type	Pin Name	MCU Extension Mode 2※10	Single Chip	Manual	Software Standby	Sleep	Bus Mastership Release※10	Oscillation Stop Detected	POE Function Used
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※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※7	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※6	Z
	PF0 to PF15	Z		I	Z	I	I	I	I

[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 MZIZDL bit in the high-current port control register (HCPCR) is cleared to 0.
6. Becomes high-impedance when the MZIZEH bit in the high-current port control register (HCPCR) is cleared to 0.
7. Becomes high-impedance when the MZIZEL bit in the high-current port control register (HCPCR) is cleared to 0.
8. Becomes input during a power-on reset. Pull-up to prevent erroneous operation. Pull-down with a resistance of at least 1 MΩ as required.
9. Pulled-up inside the LSI when there is no input.
10. SH7239A and SH7237A only.

B. Package Dimensions

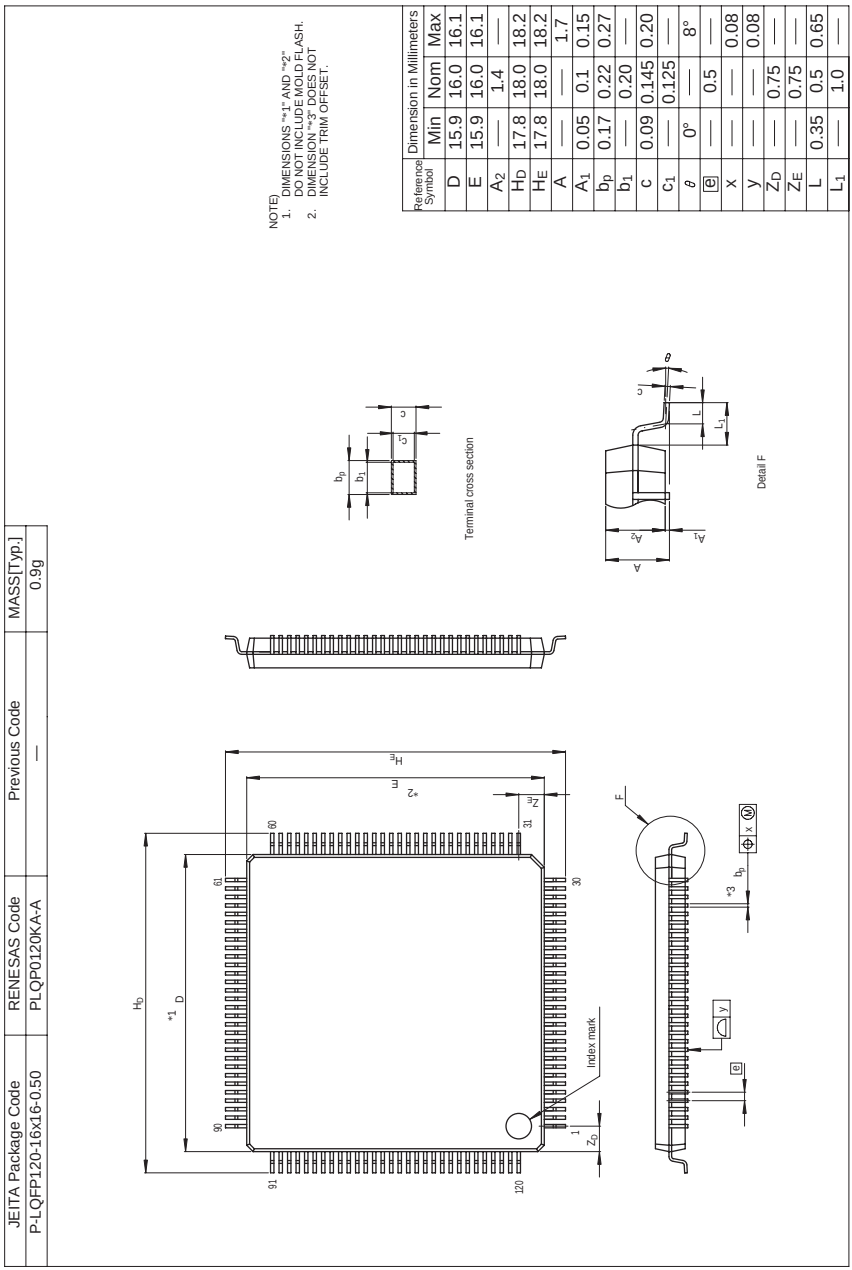


Figure B.1 Package Dimensions

Index

Numerics

16-bit/32-bit displacement 35

A

A/D conversion time 930
A/D converter (ADC) 895
A/D converter activation 542
A/D converter activation by MTU2 and MTU2S 931
A/D converter characteristics 1418
A/D converter start request delaying function 525
A/D trigger input timing 1413
Absolute accuracy 946
Absolute address 35
Absolute address accessing 35
Absolute maximum ratings 1377
AC characteristics 1385
AC characteristics measurement conditions 1417
Access size and data alignment 274
Access wait control 281
Address errors 109
Address map 247
Addressing modes 36
Arithmetic operation instructions 56
Auto-request mode 333

B

Banked register and input/output of banks 158
Bit manipulation instructions 68
Block transfer mode 225
Boot mode 1143, 1242
Branch instructions 61
Break detection and processing 738, 801
Break on data access cycle 193
Break on instruction fetch cycle 192
Burst mode 346
Bus arbitration 295
Bus connections in RAM 1266
Bus state controller (BSC) 245
Bus timing 1393

C

Calculating exception handling vector table addresses 104
CAN interface 958
CAN sleep mode 1001
Cascaded operation 459
Caution on period setting 558
Chain transfer 226
Clock frequency control circuit 79
Clock operating modes 82
Clock pulse generator (CPG) 77
Clock synchronous mode 671, 719
Clock timing 1386
CMCNT count timing 651
Compare match timer (CMT) 645
Complementary PWM mode 479
Conflict between byte-write and count-up processes of CMCNT 656
Conflict between NMI Interrupt and DTC Activation 243
Conflict between word-write and count-up processes of CMCNT 655
Conflict between write and compare-match processes of CMCNT 654
Continuous scan mode 923
Control signal timing 1389

Controller area network (RCAN-ET) 955
Controller area network timing..... 1412
Controlling RSPI pins..... 841
CPU 21
Crystal oscillator..... 79
$\overline{\text{CS}}$n assert period expansion..... 283
Cycle steal mode..... 344

D

Data format..... 21
Data format in registers 30
Data formats in memory 30
Data transfer controller (DTC) 199
Data transfer instructions..... 52
DC characteristics..... 1378
Dead time compensation 535
Definition of time quanta..... 976
Definitions of
A/D conversion accuracy..... 946
Delayed branch instructions 33
Direct memory access controller
(DMAC) 303
Displacement accessing..... 36
Divider..... 79
DMA transfer flowchart 332
DMAC and DTC activation..... 541
DMAC module timing..... 1401
DREQ pin sampling timing 349
DTC activation by interrupt..... 238
DTC activation sources 211
DTC bus release timing 234
DTC execution status..... 231
DTC vector address 213
Dual address mode..... 341

E

Effective address calculation 36
Electrical characteristics 1377
Endian 274
Equation for getting SCBRR value 763
Error protection..... 1211, 1259
Error protection types 1211
Error Protection Types..... 1260
Exception handling 99
Exception handling state 70
Exception handling vector table..... 103
Exception source generation immediately
after delayed branch instruction..... 119
Exceptions triggered by instructions..... 115
External pulse width measurement 534
External request mode..... 333
External trigger input timing..... 932

F

FCU command list 1177
FCU command usage..... 1184, 1253
Fixed mode 337
Floating-point operation instructions 65
Floating-point registers 25
Floating-point system registers 26
FPU-related CPU instructions..... 67
Full-scale error..... 946

G

General illegal instructions 117
General registers 21
Global base register (GBR)..... 24

H

Halt mode..... 1001
Hardware protection 1209, 1258
H-UDI commands..... 1292
H-UDI interrupt 136, 1295
H-UDI related pin timing..... 1415
H-UDI reset 1295

I

I/O port timing 1414
I/O ports 1085
ID Reorder 968
Immediate data 34
Immediate data accessing 34
Immediate data format 31
Initial values of control registers 29
Initial values of floating-point registers 29
Initial values of floating-point system registers 29
Initial values of general registers 29
Initial values of system registers 29
Initializing RSPI 868
Input sampling and A/D conversion time 929
Instruction features 32
Instruction format 41
Instruction set 45
Integer division instructions 117
Interrupt controller (INTC) 123
Interrupt exception handling 114
Interrupt exception handling vectors and priorities 140
Interrupt priority level 113
Interrupt requests 162
Interrupt response time 151
IRQ interrupts 137

J

Jump table base register (TBR) 24

L

Load-store architecture 32
Local acceptance filter mask (LAFM) 966
Location of transfer information and DTC vector table 211
Logic operation instructions 59
Loopback mode 890

M

Mailbox 958
Mailbox control 958
Mailbox structure 961
Manual reset 107, 1274
Master mode operation 874
MCU operating modes 71
Message control field 962
Message data fields 967
Message receive sequence 1008
Message transmission sequence 1005
Micro processor interface (MPI) 958
Module standby function 1288
Module standby mode setting 241, 740
MOSI signal value determination during SSL negate period 842
MPX-I/O interface 284
MTU2 functions 359
MTU2 interrupts 540
MTU2 output pin initialization 573
MTU2, MTU2S module timing 1402
MTU2–MTU2S synchronous operation 530
MTU2S functions 606
Multi-function timer pulse unit 2 (MTU2) 357
Multi-function timer pulse unit 2S (MTU2S) 605
Multiply and accumulate register high (MACH) 25
Multiply and accumulate register low (MACL) 25
Multiply/multiply-and-accumulate operations 33
Multiprocessor communication function 728

N

NMI interrupt 136
Nonlinearity error 946
Normal space interface 277
Normal transfer mode 222
Note on changing operating mode 75
Note on using an external crystal resonator 96
Notes on board design 948
Notes on noise countermeasures 949

O

Offset error 946
On-chip peripheral module interrupts 138
On-chip peripheral module request 335
On-chip RAM address space 1267
Operation in asynchronous mode 780
Operation in clocked synchronous mode 790
Output load circuit 1417

P

Page conflict 1272
Pin function controller (PFC) 1017
Pin states of this LSI in each processing state 1423
PLL circuit 79
POE2 interrupt source 642
POE2 module timing 1403
Port output enable 2 (POE2) 613
Power-down modes 1273
Power-down state 70
Power-on reset 1274
Procedure register (PR) 25
Program counter (PC) 25
Program execution state 70
Programming/erasing host command wait state 1165
Protection 1209, 1258

Q

Quantization error 946

R

RAM 1265
RAM block diagram 1266
RCAN-ET bit rate calculation 979
RCAN-ET interrupt sources 1012
RCAN-ET memory map 959
RCAN-ET reset sequence 1000
Receive data sampling timing and receive margin (asynchronous mode) 802
Reconfiguration of Mailbox 1010
Register addresses (by functional module, in order of the corresponding section numbers) 1298
Register bank error exception handling 111, 161
Register bank errors 111
Register bank exception 161
Register banks 29, 157
Register bits 1323
Register states in each operating mode 1356
Registers
- ABACK0 993
- ACLKCR 90
- ADANSR_0 to ADANSR_2 907
- ADBYPSCR_0 to ADBYPSCR_2 908
- ADCR_0 to ADCR_2 901
- ADDR0 to ADDR15 910
- ADDR0GR0_0 to ADDR0GR0_2 ..... 917, 918
- ADSR_0 to ADSR_2 904
- ADSTRGR_0 to ADSTRGR_2 905

ADTSR_0 to ADTSR_2 912
BAMR 172, 176, 180, 184
BAR 171, 175, 179, 183
BBR 173, 177, 181, 185
BCR0, BCR1 976
BRCR 187
BSCEHR 211, 271
CHCR 314
CMCNT 650
CMCOR 650
CMCSR 648
CMNCR 250
CMSTR 647
CRA 206
CRB 207
CSnBCR (n = 0, 1, 3 to 6) 253
CSnWCR (n = 0, 1, 3 to 6) 258
DAR (DMAC) 312
DAR (DTC) 205
DMAOR 325
DMARS0 to DMARS3 329
DMATCR 313
DTCCR 209
DTCERA to DTCERE 208
DTCVBR 210
EEPBCCNT 1238
EEPBCSTAT 1239
EEPRE0 1232
EEPRE1 1233
EEPWE0 1234
EEPWE1 1235
FAEINT 1124, 1230
FASTAT 1122, 1227
FCMDR 1136
FCPSR 1137
FCURAME 1126
FENTRYR 1132, 1236
FMODR 1121, 1226
FPESTAT 1138
FPMON 1120
FPROTR 1134
FPSCR 27
FPUL 26
FRESETR 1135
FRQCR 86
FSTATR0 1127
FSTATR1 1131
GSR 973
HCPCR 1074
IBCR 133
IBNR 134
ICR0 129
ICR1 130
ICSR1 618
ICSR2 621
ICSR3 624
IMR 986
IPR01, IPR02, IPR05 to IPR18 127
IRQRR 131
IRR 981
MBIMR0 996
MCLKCR 89
MCR 968
MRA 202
MRB 203
OCSR1 620
OCSR2 622
OSCCR 91
PACRH1 1026
PACRL1 1031
PACRL2 1030
PACRL3 1029
PACRL4 1028
PADRH 1087
PADRL 1087
PAIORH 1025
PAIORL 1025
PAPCRH 1032
PAPCRL 1033
PAPRH 1089

PAPRL 1089
PBCRH1 1037
PBCRH2 1035
PBCRL1 1040
PBCRL2 1039
PBDRH 1092
PBDRL 1092
PBIORH 1034
PBIORL 1034
PBPCRH 1042
PBPCRL 1043
PBPRH 1094
PBPRL 1095
PCCRL1 1051
PCCRL2 1049
PCCRL3 1047
PCCRL4 1044
PCDRL 1097
PCIORL 1044
PCKAR 1140
PCPCRL 1053
PCPRL 1099
PDCRL1 1061
PDCRL2 1059
PDCRL3 1057
PDCRL4 1055
PDDRL 1101
PDIORL 1054
PDPCRL 1063
PDPRL 1103
PECRL1 1070
PECRL2 1068
PECRL3 1067
PECRL4 1065
PEDRL 1106
PEIORL 1064
PEPCRL 1073
PEPRL 1107
PFDRL 1109
POECR1 627
POECR2 629
POECR3 634
RCCR 1139
RDAR 323
RDMATCR 324
REC 986
RFPR0 995
ROMMAT 1125
RSAR 322
RXPR0 994
SAR (DMAC) 311
SAR (DTC) 204
SCBRR (SCI) 693
SCBRR (SCIF) 763
SCFCR 771
SCFDR 773
SCFRDR 746
SCFSR 755
SCFTDR 747
SCLSR 775
SCRDR (SCI) 676
SCRSR (SCI) 676
SCRSR (SCIF) 746
SCSCR (SCI) 681
SCSCR (SCIF) 751
SCSDCR 692
SCSEMR 777
SCSMR (SCI) 677
SCSMR (SCIF) 748
SCSPTR (SCI) 690
SCSPTR (SCIF) 774
SCSSR 684
SCTDR (SCI) 677
SCTSR (SCI) 676
SCTSR (SCIF) 747
SDBPR 1291
SDIR 1292
SPBR 825
SPCKD 831
SPCMD 834

SPCR 811
SPDR 821
SPND 833
SPOER 626
SPPCR 815
SPSCR 822
SPSR 816
SPSSR 823
SSLND 832
SSLP 814
STBCR 1276
STBCR2 1277
STBCR3 1278
STBCR4 1280
STBCR5 1281
STBCR6 1283
SYSCR1 1268
SYSCR2 1270
TADCOBRA_4 416
TADCOBRB_4 416
TADCORA_4 416
TADCORB_4 416
TADCR 413
TBTER 441
TBTM 408
TCBR 438
TCDR 437
TCNT 417
TCNTCMPCLR 395
TCNTS 436
TCR 369
TCSYSTR 422
TDDR 437
TDER 443
TEC 986
TGCR 434
TGR 417
TICCR 410
TIER 396
TIOR 376
TITCNT 440
TITCR 438
TMDR 373
TOCR1 427
TOCR2 430
TOER 426
TOLBR 433
TRWER 425
TSR 401
TSTR 418
TSYCRS 411
TSYR 420
TWCR 444
TXACK0 992
TXCR0 991
TXPR1, TXPR0 989
UMSR0 997
WRCSR 662
WTCNT 659
WTCSR 660
Relationship between RSPI modes and SPCR and description of each mode 839
Renesas serial peripheral interface (RSPI) 805
Repeat transfer mode 223
Reset configuration 1294
Reset state 70
Reset-synchronized PWM mode 476
Restoration from bank 159
Restoration from stack 160
Restriction on DMAC and DTC usage 801
RISC-type instruction set 32
ROM 1113
Round-robin mode 337
RSPI data format 855
RSPI error detection function 863
RSPI system configuration example 843
RSPI transfer format 853, 854

S

Saving to bank 158
Saving to stack 160
SCI interrupt sources 734
SCIF interrupt sources 799
SCIF module timing 1406
SCSPTR and SCI pins 735
Sending a break signal 738, 801
Serial communication interface (SCI) 671
Serial communication interface with FIFO (SCIF) 743
Shift instructions 60
Sign extension of word data 32
Single address mode 343
Single chip mode 72
Single-cycle scan mode 920
Slave mode operation 881
Sleep mode 1284
Slot illegal instructions 116
Software protection 1210, 1258
Software standby mode 1284
Stack after interrupt exception handling 150
Stack status after exception handling ends 120
Standby control circuit 79
State transition in boot mode 1144
Status register (SR) 23
Supported DMA transfers 340
Suspending operation 1203
System control instructions 63

T

T bit 33
TAP controller 1293
TDO output timing 1294
Test mode settings 998
The address map for each mailbox 960
The address map for the operating modes 73
Timing to clear an interrupt source 165
Transfer information read skip function 221
Transfer information writeback skip function 222
Transmission buffer empty/ receive buffer full flags 861
Trap instructions 116
Types of exception handling and priority order 99

U

UBC trigger timing 1400
Unconditional branch instructions with no delay slot 33
User boot mode 1206
User break controller (UBC) 167
User break interrupt 136
User debugging interface (H-UDI) 1289
User program mode 1177
Using interval timer mode 667
Using watchdog timer mode 665

V

Vector base register (VBR) 24

W

Wait between access cycles 288
Watchdog timer (WDT) 657
Watchdog timer timing 1404

SH7239 Group, SH7237 Group User's Manual: Hardware

Publication Date: Rev.1.00 Sep 13, 2010

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, 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.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0

SH7239 Group, SH7237 Group

Renesas Electronics Corporation

R01UH0086EJ0100