REJ09B0344-0200

Everywhere you imagine.

The revision list can be viewed directly by clicking the title page. The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

32

SH7211 Group

Hardware Manual

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family

SH7211 R5F72115D160FPV

Hardware Manual

Rev.2.00
Revision Date: May. 8, 2008

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test registers may have been be allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

     The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. List of Registers
8. Electrical Characteristics
9. Appendix
   - Product Type, Package Dimensions, etc.
10. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

11. Index

Preface

This LSI is an RISC (Reduced Instruction Set Computer) microcomputer which includes a Renesas Technology-original RISC CPU as its core, and the peripheral functions required to configure a system.

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users.
Refer to the SH-2A, SH2A-FPU Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the SH-2A, SH2A-FPU Software Manual.
- In order to understand the details of a register when its name is known
  Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 26, List of Registers.

- Examples

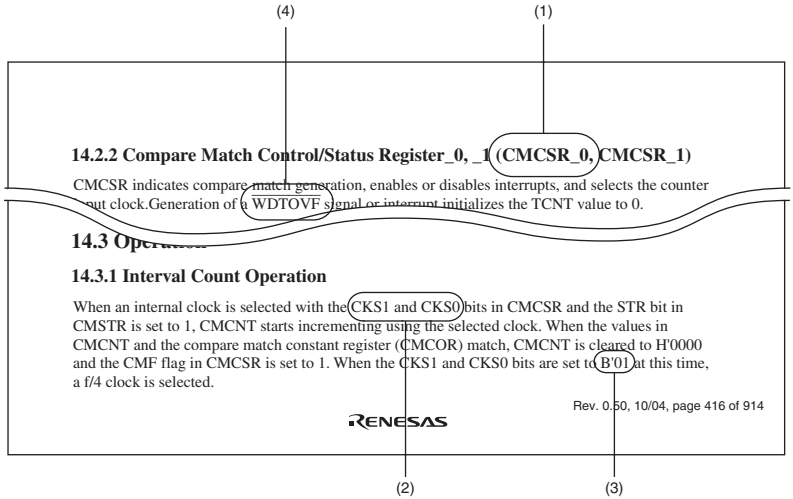
  The notation used for register names, bit names, numbers, and symbols in this manual is described below.

(1) Registers
The style (register name)_(channel number) is used in cases where the same or a similar function is implemented on more than one channel.
Example: CMCSR_0

(2) Bits
When bit names are given in this manual, the higher-order bits are to the left and the lower-order bits are to the right.
Example: CKS1, CKS0

(3) Numbers
Binary numbers are given as B'xxxx, hexadecimal are given as H'xxxx, and decimal are given as xxxx.
Examples: B'11 or 11, H'EFA0, 1234

(4) Symbols
An overbar is added to the names of active-low signals.
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.

- Notation in bit figures and tables describing arrangements of bits

  Each register description includes a figure that illustrates the arrangement of bits and a table that describes the meanings of settings in the bits.

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

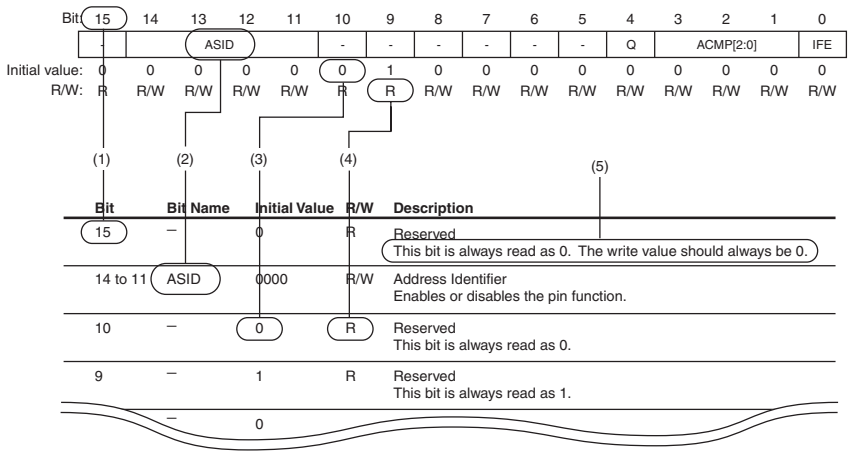
(2) Bit Name
The short form of the name of the bit or bit field within the register.
When the individual bits of bit fields have to be clearly indicated, notation allowing this is included (e.g., ASID[3:0]).
A reserved bit is indicated by –.
Instead of a bit name, a blank is used for some bits, such as those of timer counters.

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

(4) R/W
Indicates whether each bit is readable or writable, or either writing to or reading from the bit is prohibited.
The notation is as follows:

R/W: Bit or field is readable and writable.
R/(W): Bit or field is readable and writable.
However, writing is only performed to clear the flag.
R: Bit or field is readable and writable.
However, "R" is indicated for all reserved bits. When writing to the bit is required, write the value stated in the bit table or the initial value.
W: Bit or field is readable and writable.
However, only the value in the bit table is guaranteed when reading from the bit.

(5) Description
Describes the function enabled by setting the bit.



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

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

Contents

Section 1 Overview 1
1.1 SH7211 Features 1
1.2 Block Diagram 7
1.3 Pin Arrangement 8
1.4 Pin Functions 9

Section 2 CPU 15
2.1 Register Configuration 15
2.1.1 General Registers 15
2.1.2 Control Registers 16
2.1.3 System Registers 18
2.1.4 Register Banks 19
2.1.5 Initial Values of Registers 19
2.2 Data Formats 20
2.2.1 Data Format in Registers 20
2.2.2 Data Formats in Memory 20
2.2.3 Immediate Data Format 21
2.3 Instruction Features 22
2.3.1 RISC-Type Instruction Set 22
2.3.2 Addressing Modes 26
2.3.3 Instruction Format 31
2.4 Instruction Set 35
2.4.1 Instruction Set by Classification 35
2.4.2 Data Transfer Instructions 40
2.4.3 Arithmetic Operation Instructions 44
2.4.4 Logic Operation Instructions 47
2.4.5 Shift Instructions 48
2.4.6 Branch Instructions 49
2.4.7 System Control Instructions 50
2.4.8 Bit Manipulation Instructions 52
2.5 Processing States 53

Section 3 MCU Operating Modes 55
3.1 Selection of Operating Modes 55
3.2 Input/Output Pins 56
3.3 Operating Modes 57

3.3.1 Mode 0 (MCU Extension Mode 0) 57
3.3.2 Mode 1 (MCU Extension Mode 1) 57
3.3.3 Mode 2 (MCU Extension Mode 2) 57
3.3.4 Mode 3 (Single Chip Mode) 57
3.4 Address Map 58
3.5 Initial State in This LSI 59
3.6 Note on Changing Operating Mode 59

Section 4 Clock Pulse Generator (CPG) 61
4.1 Features 61
4.2 Input/Output Pins 65
4.3 Clock Operating Modes 66
4.4 Register Descriptions 68
4.4.1 Frequency Control Register (FRQCR) 68
4.4.2 MTU2S Clock Frequency Control Register (MCLKCR) 71
4.4.3 AD Clock Frequency Control Register (ACLKCR) 72
4.5 Changing the Frequency 73
4.5.1 Changing the Multiplication Rate 73
4.5.2 Changing the Division Ratio 74
4.6 Notes on Board Design 75
4.6.1 Note on Using an External Crystal Resonator 75
4.6.2 Note on Bypass Capacitor 75
4.6.3 Note on Using a PLL Oscillation Circuit 75

Section 5 Exception Handling 77
5.1 Overview 77
5.1.1 Types of Exception Handling and Priority 77
5.1.2 Exception Handling Operations 79
5.1.3 Exception Handling Vector Table 81
5.2 Resets 83
5.2.1 Types of Reset 83
5.2.2 Power-On Reset 84
5.2.3 Manual Reset 86
5.3 Address Errors 87
5.3.1 Address Error Sources 87
5.3.2 Address Error Exception Handling 88
5.4 Register Bank Errors 89
5.4.1 Register Bank Error Sources 89
5.4.2 Register Bank Error Exception Handling 89
5.5 Interrupts 90

5.5.1 Interrupt Sources ... 90
5.5.2 Interrupt Priority Level ... 91
5.5.3 Interrupt Exception Handling ... 92
5.6 Exceptions Triggered by Instructions ... 93
5.6.1 Types of Exceptions Triggered by Instructions ... 93
5.6.2 Trap Instructions ... 94
5.6.3 Slot Illegal Instructions ... 94
5.6.4 General Illegal Instructions ... 95
5.6.5 Integer Division Instructions ... 95
5.7 When Exception Sources Are Not Accepted ... 96
5.8 Stack Status after Exception Handling Ends ... 97
5.9 Usage Notes ... 99
5.9.1 Value of Stack Pointer (SP) ... 99
5.9.2 Value of Vector Base Register (VBR) ... 99
5.9.3 Address Errors Caused by Stacking of Address Error Exception Handling ... 99

Section 6 Interrupt Controller (INTC) ... 101
6.1 Features ... 101
6.2 Input/Output Pins ... 103
6.3 Register Descriptions ... 104
6.3.1 Interrupt Priority Registers 01, 02, 05 to 15 (IPR01, IPR02, IPR05 to IPR15) ... 105
6.3.2 Interrupt Control Register 0 (ICR0) ... 107
6.3.3 Interrupt Control Register 1 (ICR1) ... 108
6.3.4 IRQ Interrupt Request Register (IRQRR) ... 109
6.3.5 Bank Control Register (IBCR) ... 111
6.3.6 Bank Number Register (IBNR) ... 112
6.4 Interrupt Sources ... 114
6.4.1 NMI Interrupt ... 114
6.4.2 User Break Interrupt ... 114
6.4.3 H-UDI Interrupt ... 114
6.4.4 IRQ Interrupts ... 115
6.4.5 On-Chip Peripheral Module Interrupts ... 116
6.5 Interrupt Exception Handling Vector Table and Priority ... 117
6.6 Operation ... 125
6.6.1 Interrupt Operation Sequence ... 125
6.6.2 Stack after Interrupt Exception Handling ... 128
6.7 Interrupt Response Time ... 129
6.8 Register Banks ... 135
6.8.1 Banked Register and Input/Output of Banks ... 136
6.8.2 Bank Save and Restore Operations ... 136

6.8.3 Save and Restore Operations after Saving to All Banks 138
6.8.4 Register Bank Exception 139
6.8.5 Register Bank Error Exception Handling 139
6.9 Data Transfer with Interrupt Request Signals 140
6.9.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but Not DMAC Activating 141
6.9.2 Handling Interrupt Request Signals as Sources for Activating DMAC but Not CPU Interrupt 141
6.10 Usage Note 141
6.10.1 Timing to Clear an Interrupt Source 141

Section 7 User Break Controller (UBC) 143
7.1 Features 143
7.2 Input/Output Pin 145
7.3 Register Descriptions 145
7.3.1 Break Address Register_0 (BAR_0) 146
7.3.2 Break Address Mask Register_0 (BAMR_0) 147
7.3.3 Break Bus Cycle Register_0 (BBR_0) 148
7.3.4 Break Address Register_1 (BAR_1) 150
7.3.5 Break Address Mask Register_1 (BAMR_1) 151
7.3.6 Break Bus Cycle Register_1 (BBR_1) 152
7.3.7 Break Address Register_2 (BAR_2) 154
7.3.8 Break Address Mask Register_2 (BAMR_2) 155
7.3.9 Break Bus Cycle Register_2 (BBR_2) 156
7.3.10 Break Address Register_3 (BAR_3) 158
7.3.11 Break Address Mask Register_3 (BAMR_3) 159
7.3.12 Break Bus Cycle Register_3 (BBR_3) 160
7.3.13 Break Control Register (BRCR) 162
7.4 Operation 166
7.4.1 Flow of the User Break Operation 166
7.4.2 Break on Instruction Fetch Cycle 167
7.4.3 Break on Data Access Cycle 168
7.4.4 Value of Saved Program Counter 169
7.4.5 Usage Examples 170
7.5 Usage Notes 173

Section 8 Bus State Controller (BSC) 175
8.1 Features 175
8.2 Input/Output Pins 178
8.3 Area Overview 179

8.3.1 Address Map 179
8.3.2 Setting Operating Modes 180
8.4 Register Descriptions 182
8.4.1 Common Control Register (CMNCR) 183
8.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7) 186
8.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7) 191
8.4.4 SDRAM Control Register (SDCR) 221
8.4.5 Refresh Timer Control/Status Register (RTCSR) 225
8.4.6 Refresh Timer Counter (RTCNT) 227
8.4.7 Refresh Time Constant Register (RTCOR) 228
8.5 Operation 229
8.5.1 Endian/Access Size and Data Alignment 229
8.5.2 Normal Space Interface 232
8.5.3 Access Wait Control 236
8.5.4 $\overline{\text{CSn}}$ Assert Period Expansion 238
8.5.5 MPX-I/O Interface 239
8.5.6 SDRAM Interface 243
8.5.7 Burst ROM (Clock Asynchronous) Interface 278
8.5.8 SRAM Interface with Byte Selection 280
8.5.9 Burst ROM (Clock Synchronous) Interface 285
8.5.10 Wait between Access Cycles 286
8.5.11 Bus Arbitration 293
8.5.12 Others 295
8.6 Usage Notes 298
8.6.1 Burst ROM Interface 298

Section 9 Direct Memory Access Controller (DMAC) 299
9.1 Features 299
9.2 Input/Output Pins 302
9.3 Register Descriptions 303
9.3.1 DMA Source Address Registers (SAR) 307
9.3.2 DMA Destination Address Registers (DAR) 308
9.3.3 DMA Transfer Count Registers (DMATCR) 309
9.3.4 DMA Channel Control Registers (CHCR) 310
9.3.5 DMA Reload Source Address Registers (RSAR) 318
9.3.6 DMA Reload Destination Address Registers (RDAR) 319
9.3.7 DMA Reload Transfer Count Registers (RDMATCR) 320
9.3.8 DMA Operation Register (DMAOR) 321
9.3.9 DMA Extension Resource Selectors 0 to 3 (DMARS0 to DMARS3) 325
9.4 Operation 327

9.4.1 Transfer Flow 327
9.4.2 DMA Transfer Requests 329
9.4.3 Channel Priority 333
9.4.4 DMA Transfer Types 336
9.4.5 Number of Bus Cycles and DREQ Pin Sampling Timing 345
9.5 Usage Note 349
9.5.1 Half-End Flag Setting and Half-End Interrupt 349

Section 10 Multi-Function Timer Pulse Unit 2 (MTU2) 351
10.1 Features 351
10.2 Input/Output Pins 357
10.3 Register Descriptions 358
10.3.1 Timer Control Register (TCR) 362
10.3.2 Timer Mode Register (TMDR) 366
10.3.3 Timer I/O Control Register (TIOR) 369
10.3.4 Timer Compare Match Clear Register (TCNTCMPCLR) 388
10.3.5 Timer Interrupt Enable Register (TIER) 389
10.3.6 Timer Status Register (TSR) 394
10.3.7 Timer Buffer Operation Transfer Mode Register (TBTM) 401
10.3.8 Timer Input Capture Control Register (TICCR) 403
10.3.9 Timer Synchronous Clear Register (TSYCR) 404
10.3.10 Timer A/D Converter Start Request Control Register (TADCR) 406
10.3.11 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) 409
10.3.12 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) 409
10.3.13 Timer Counter (TCNT) 410
10.3.14 Timer General Register (TGR) 410
10.3.15 Timer Start Register (TSTR) 411
10.3.16 Timer Synchronous Register (TSYR) 413
10.3.17 Timer Counter Synchronous Start Register (TCSYSTR) 415
10.3.18 Timer Read/Write Enable Register (TRWER) 418
10.3.19 Timer Output Master Enable Register (TOER) 419
10.3.20 Timer Output Control Register 1 (TOCR1) 420
10.3.21 Timer Output Control Register 2 (TOCR2) 423
10.3.22 Timer Output Level Buffer Register (TOLBR) 426
10.3.23 Timer Gate Control Register (TGCR) 427
10.3.24 Timer Subcounter (TCNTS) 429
10.3.25 Timer Dead Time Data Register (TDDR) 430
10.3.26 Timer Cycle Data Register (TCDR) 430

10.3.27 Timer Cycle Buffer Register (TCBR) 431
10.3.28 Timer Interrupt Skipping Set Register (TITCR) 431
10.3.29 Timer Interrupt Skipping Counter (TITCNT) 433
10.3.30 Timer Buffer Transfer Set Register (TBTER) 434
10.3.31 Timer Dead Time Enable Register (TDER) 436
10.3.32 Timer Waveform Control Register (TWCR) 437
10.3.33 Bus Master Interface 439
10.4 Operation 440
10.4.1 Basic Functions 440
10.4.2 Synchronous Operation 446
10.4.3 Buffer Operation 448
10.4.4 Cascaded Operation 452
10.4.5 PWM Modes 457
10.4.6 Phase Counting Mode 462
10.4.7 Reset-Synchronized PWM Mode 469
10.4.8 Complementary PWM Mode 472
10.4.9 A/D Converter Start Request Delaying Function 518
10.4.10 MTU2–MTU2S Synchronous Operation 522
10.4.11 External Pulse Width Measurement 528
10.4.12 Dead Time Compensation 529
10.4.13 TCNT Capture at Crest and/or Trough in Complementary PWM Operation 532
10.5 Interrupt Sources 533
10.5.1 Interrupt Sources and Priorities 533
10.5.2 DMAC Activation 535
10.5.3 A/D Converter Activation 536
10.6 Operation Timing 538
10.6.1 Input/Output Timing 538
10.6.2 Interrupt Signal Timing 545
10.7 Usage Notes 550
10.7.1 Module Standby Mode Setting 550
10.7.2 Input Clock Restrictions 550
10.7.3 Caution on Period Setting 551
10.7.4 Contention between TCNT Write and Clear Operations 551
10.7.5 Contention between TCNT Write and Increment Operations 552
10.7.6 Contention between TGR Write and Compare Match 553
10.7.7 Contention between Buffer Register Write and Compare Match 554
10.7.8 Contention between Buffer Register Write and TCNT Clear 555
10.7.9 Contention between TGR Read and Input Capture 556
10.7.10 Contention between TGR Write and Input Capture 557
10.7.11 Contention between Buffer Register Write and Input Capture 558

10.7.12 TCNT2 Write and Overflow/Underflow Contention in Cascade Connection 558
10.7.13 Counter Value during Complementary PWM Mode Stop 560
10.7.14 Buffer Operation Setting in Complementary PWM Mode 560
10.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag 561
10.7.16 Overflow Flags in Reset Synchronous PWM Mode 562
10.7.17 Contention between Overflow/Underflow and Counter Clearing 563
10.7.18 Contention between TCNT Write and Overflow/Underflow 564
10.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode 564
10.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 565
10.7.21 Interrupts in Module Standby Mode 565
10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection 565
10.8 MTU2 Output Pin Initialization 566
10.8.1 Operating Modes 566
10.8.2 Reset Start Operation 566
10.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc. 567
10.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. 568

Section 11 Multi-Function Timer Pulse Unit 2S (MTU2S) 599
11.1 Input/Output Pins 603
11.2 Register Descriptions 604

Section 12 Port Output Enable 2 (POE2) 607
12.1 Features 608
12.2 Input/Output Pins 610
12.3 Register Descriptions 612
12.3.1 Input Level Control/Status Register 1 (ICSR1) 613
12.3.2 Output Level Control/Status Register 1 (OCSR1) 617
12.3.3 Input Level Control/Status Register 2 (ICSR2) 618
12.3.4 Output Level Control/Status Register 2 (OCSR2) 621
12.3.5 Input Level Control/Status Register 3 (ICSR3) 622
12.3.6 Software Port Output Enable Register (SPOER) 624
12.3.7 Port Output Enable Control Register 1 (POECR1) 626
12.3.8 Port Output Enable Control Register 2 (POECR2) 627
12.4 Operation 630
12.4.1 Input Level Detection Operation 631
12.4.2 Output-Level Compare Operation 632
12.4.3 Release from High-Impedance State 633

12.5 Interrupts 634
12.6 Usage Note 635
12.6.1 Pin Status When the WDT Issues a Power-On Reset 635

Section 13 Compare Match Timer (CMT) 637
13.1 Features 637
13.2 Register Descriptions 638
13.2.1 Compare Match Timer Start Register (CMSTR) 639
13.2.2 Compare Match Timer Control/Status Register (CMCSR) 640
13.2.3 Compare Match Counter (CMCNT) 642
13.2.4 Compare Match Constant Register (CMCOR) 642
13.3 Operation 643
13.3.1 Interval Count Operation 643
13.3.2 CMCNT Count Timing 643
13.4 Interrupts 644
13.4.1 Interrupt Sources and DMA Transfer Requests 644
13.4.2 Timing of Compare Match Flag Setting 644
13.4.3 Timing of Compare Match Flag Clearing 645
13.5 Usage Notes 646
13.5.1 Conflict between Write and Compare-Match Processes of CMCNT 646
13.5.2 Conflict between Word-Write and Count-Up Processes of CMCNT 647
13.5.3 Conflict between Byte-Write and Count-Up Processes of CMCNT 648
13.5.4 Compare Match Between CMCNT and CMCOR 648

Section 14 Watchdog Timer (WDT) 649
14.1 Features 649
14.2 Input/Output Pin 651
14.3 Register Descriptions 652
14.3.1 Watchdog Timer Counter (WTCNT) 652
14.3.2 Watchdog Timer Control/Status Register (WTCSR) 653
14.3.3 Watchdog Reset Control/Status Register (WRCSR) 655
14.3.4 Notes on Register Access 656
14.4 WDT Usage 658
14.4.1 Canceling Software Standby Mode 658
14.4.2 Changing the Frequency 659
14.4.3 Using Watchdog Timer Mode 660
14.4.4 Using Interval Timer Mode 662
14.5 Usage Notes 663
14.5.1 Timer Variation 663
14.5.2 Prohibition against Setting H'FF to WTCNT 663

14.5.3 System Reset by $\overline{\text{WDTOVF}}$ Signal 663
14.5.4 Manual Reset in Watchdog Timer Mode 664

Section 15 Serial Communication Interface with FIFO (SCIF) 665
15.1 Features 665
15.2 Input/Output Pins 667
15.3 Register Descriptions 668
15.3.1 Receive Shift Register (SCRSR) 670
15.3.2 Receive FIFO Data Register (SCFRDR) 670
15.3.3 Transmit Shift Register (SCTSR) 671
15.3.4 Transmit FIFO Data Register (SCFTDR) 671
15.3.5 Serial Mode Register (SCSMR) 672
15.3.6 Serial Control Register (SCSCR) 675
15.3.7 Serial Status Register (SCFSR) 679
15.3.8 Bit Rate Register (SCBRR) 687
15.3.9 FIFO Control Register (SCFCR) 692
15.3.10 FIFO Data Count Register (SCFDR) 694
15.3.11 Serial Port Register (SCSPTR) 695
15.3.12 Line Status Register (SCLSR) 696
15.3.13 Serial Extended Mode Register (SCSEMR) 698
15.4 Operation 699
15.4.1 Overview 699
15.4.2 Operation in Asynchronous Mode 701
15.4.3 Operation in Clocked Synchronous Mode 711
15.5 SCIF Interrupts 720
15.6 Usage Notes 721
15.6.1 SCFTDR Writing and TDFE Flag 721
15.6.2 SCFRDR Reading and RDF Flag 721
15.6.3 Restriction on DMAC Usage 722
15.6.4 Break Detection and Processing 722
15.6.5 Sending a Break Signal 722
15.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) 723
15.6.7 FER and PER Flags in the Serial Status Register (SCFSR) 724

Section 16 I^2C Bus Interface 3 (IIC3) 725
16.1 Features 725
16.2 Input/Output Pins 727
16.3 Register Descriptions 728
16.3.1 I^2C Bus Control Register 1 (ICCR1) 729
16.3.2 I^2C Bus Control Register 2 (ICCR2) 732

16.3.3 I²C Bus Mode Register (ICMR) 734
16.3.4 I²C Bus Interrupt Enable Register (ICIER) 736
16.3.5 I²C Bus Status Register (ICSR) 738
16.3.6 Slave Address Register (SAR) 741
16.3.7 I²C Bus Transmit Data Register (ICDRT) 741
16.3.8 I²C Bus Receive Data Register (ICDRR) 742
16.3.9 I²C Bus Shift Register (ICDRS) 742
16.3.10 NF2CYC Register (NF2CYC) 743
16.4 Operation 744
16.4.1 I²C Bus Format 744
16.4.2 Master Transmit Operation 745
16.4.3 Master Receive Operation 747
16.4.4 Slave Transmit Operation 749
16.4.5 Slave Receive Operation 752
16.4.6 Clocked Synchronous Serial Format 754
16.4.7 Noise Filter 758
16.4.8 Example of Use 759
16.5 Interrupt Requests 763
16.6 Bit Synchronous Circuit 764
16.7 Usage Notes 767
16.7.1 Note on Multiple Master Usage 767
16.7.2 Note on Master Receive Mode 767
16.7.3 Note on Master Receive Mode with ACKBT Setting 767
16.7.4 Note on MST and TRS Bit Status When an Arbitration was Lost 767

Section 17 A/D Converter (ADC) 769
17.1 Features 769
17.2 Input/Output Pins 771
17.3 Register Descriptions 772
17.3.1 A/D Control Register (ADCR) 773
17.3.2 A/D Status Register (ADSR) 776
17.3.3 A/D Start Trigger Select Register (ADSTRGR) 777
17.3.4 A/D Analog Input Channel Select Register (ADANSR) 779
17.3.5 A/D Data Registers 0 to 7 (ADDR0 to ADDR7) 780
17.4 Operation 781
17.4.1 Single-Cycle Scan Mode 781
17.4.2 Continuous Scan Mode 783
17.4.3 Input Sampling and A/D Conversion Time 785
17.4.4 A/D Converter Activation by MTU2 and MTU2S 787
17.4.5 External Trigger Input Timing 788

17.4.6 Example of ADDR Auto-Clear Function 788
17.5 Interrupt Sources and DMAC Transfer Requests 790
17.6 Definitions of A/D Conversion Accuracy 791
17.7 Usage Notes 793
17.7.1 Analog Input Voltage Range 793
17.7.2 Relationship between AVcc, AVss and Vcc, Vss 793
17.7.3 Range of AVREF Pin Settings 793
17.7.4 Notes on Board Design 793
17.7.5 Notes on Noise Countermeasures 794
17.7.6 Notes on Register Setting 794
17.7.7 Treatment of AVcc and AVss When the A/D Converter is Not Used 794

Section 18 D/A Converter (DAC) 795
18.1 Features 795
18.2 Input/Output Pins 796
18.3 Register Descriptions 797
18.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1) 797
18.3.2 D/A Control Register (DACR) 798
18.4 Operation 800
18.5 Usage Notes 801
18.5.1 Module Standby Mode Setting 801
18.5.2 D/A Output Hold Function in Software Standby Mode 801
18.5.3 Setting Analog Input Voltage 801

Section 19 Pin Function Controller (PFC) 803
19.1 Register Descriptions 809
19.1.1 Port A I/O Registers H, L (PAIORH, PAIORL) 811
19.1.2 Port A Control Registers H1 to H3, L1 to L4 (PACRH1 to PACRH3, PACRL1 to PACRL4) 812
19.1.3 Port B I/O Registers H, L (PBIORH, PBIORL) 827
19.1.4 Port B Control Registers H1 to H4, L1 to L4 (PBCRH1 to PBCRH4, PBCRL1 to PBCRL4) 828
19.1.5 Port D I/O Register (PDIOR) 844
19.1.6 Port D Control Registers L1 to L4 (PDCRL1 to PDCRL4) 844
19.1.7 Port F Control Register L1 (PFCRL1) 857
19.1.8 IRQOUT Function Control Register (IFCR) 859
19.1.9 WAVE Function Control Registers 1, 2 (WAVECR1, WAVECR2) 860

Section 20 I/O Ports 863
20.1 Port A 863

20.1.1 Register Descriptions 864
20.1.2 Port A Data Registers H, L (PADRH, PADRL) 864
20.1.3 Port A Port Registers H, L (PAPRH, PAPRL) 868
20.2 Port B 870
20.2.1 Register Descriptions 871
20.2.2 Port B Data Registers H, L (PBDRH, PBDRL) 871
20.2.3 Port B Port Registers H, L (PBPRH, PBPRL) 875
20.3 Port D 877
20.3.1 Register Descriptions 877
20.3.2 Port D Data Register L (PDDRL) 878
20.3.3 Port D Port Register L (PDPRL) 880
20.4 Port F 881
20.4.1 Register Descriptions 881
20.4.2 Port F Data Register (PFDR) 882
20.5 Usage Note 883

Section 21 Flash Memory 885
21.1 Features 885
21.2 Overview 887
21.2.1 Block Diagram 887
21.2.2 Operating Mode 888
21.2.3 Mode Comparison 890
21.2.4 Flash Memory Configuration 891
21.2.5 Block Division 892
21.2.6 Programming/Erasing Interface 893
21.3 Input/Output Pins 895
21.4 Register Descriptions 895
21.4.1 Registers 895
21.4.2 Programming/Erasing Interface Registers 898
21.4.3 Programming/Erasing Interface Parameters 905
21.5 On-Board Programming Mode 920
21.5.1 Boot Mode 920
21.5.2 User Program Mode 924
21.5.3 User Boot Mode 935
21.6 Protection 940
21.6.1 Hardware Protection 940
21.6.2 Software Protection 941
21.6.3 Error Protection 941
21.7 Usage Notes 943
21.7.1 Switching between User MAT and User Boot MAT 943

21.7.2 Interrupts during Programming/Erasing 944
21.7.3 Other Notes 946
21.8 Supplementary Information 949
21.8.1 Specifications of the Standard Serial Communications Interface in Boot Mode 949
21.8.2 Areas for Storage of the Procedural Program and Data for Programming 979
21.9 Programmer Mode 986

Section 22 On-Chip RAM 987
22.1 Features 987
22.2 Usage Notes 988
22.2.1 Page Conflict 988
22.2.2 RAME and RAMWE Bits 988

Section 23 Power-Down Modes 991
23.1 Features 991
23.1.1 Power-Down Modes 991
23.1.2 Reset 993
23.2 Input/Output Pins 994
23.3 Register Descriptions 994
23.3.1 Standby Control Register (STBCR) 995
23.3.2 Standby Control Register 2 (STBCR2) 996
23.3.3 Standby Control Register 3 (STBCR3) 997
23.3.4 Standby Control Register 4 (STBCR4) 999
23.3.5 System Control Register 1 (SYSCR1) 1001
23.3.6 System Control Register 2 (SYSCR2) 1003
23.4 Operation 1005
23.4.1 Sleep Mode 1005
23.4.2 Software Standby Mode 1005
23.4.3 Software Standby Mode Application Example 1008
23.4.4 Module Standby Function 1009

Section 24 User Debugging Interface (H-UDI) 1011
24.1 Features 1011
24.2 Input/Output Pins 1012
24.3 Register Descriptions 1013
24.3.1 Bypass Register (SDBPR) 1013
24.3.2 Instruction Register (SDIR) 1013
24.4 Operation 1015
24.4.1 TAP Controller 1015

24.4.2 Reset Configuration 1016
24.4.3 TDO Output Timing 1016
24.4.4 H-UDI Reset 1017
24.4.5 H-UDI Interrupt 1017
24.5 Usage Notes 1018

Section 25 WAVE Interface (WAVEIF) 1019
25.1 Features 1019
25.2 Input/Output Pins 1019

Section 26 List of Registers 1021
26.1 Register Addresses
(by functional module, in order of the corresponding section numbers) 1022
26.2 Register Bits 1037
26.3 Register States in Each Operating Mode 1066

Section 27 Electrical Characteristics 1081
27.1 Absolute Maximum Ratings 1081
27.2 Power-on/Power-off Sequence 1082
27.3 DC Characteristics 1084
27.4 AC Characteristics 1088
27.4.1 Clock Timing 1089
27.4.2 Control Signal Timing 1092
27.4.3 Bus Timing 1095
27.4.4 UBC Trigger Timing 1125
27.4.5 DMAC Module Timing 1125
27.4.6 MTU2, MTU2S Module Timing 1127
27.4.7 POE2 Module Timing 1128
27.4.8 Watchdog Timer Timing 1128
27.4.9 SCIF Module Timing 1129
27.4.10 IIC3 Module Timing 1130
27.4.11 A/D Trigger Input Timing 1131
27.4.12 I/O Port Timing 1132
27.4.13 H-UDI Related Pin Timing 1133
27.4.14 AC Characteristics Measurement Conditions 1135
27.5 A/D Converter Characteristics 1136
27.6 D/A Converter Characteristics 1137
27.7 Flash Memory Characteristics 1137

Appendix 1139
A. Pin States 1139
B. Product Lineup 1144
C. Package Dimensions 1145

Main Revisions for This Edition 1147

Index 1187

Section 1 Overview

1.1 SH7211 Features

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

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

In addition, this LSI includes on-chip peripheral functions necessary for system configuration, such as a large-capacity ROM, a ROM cache, a RAM, a direct memory access controller (DMAC), multi-function timer pulse units 2 (MTU2 and MTU2S), a serial communication interface with FIFO (SCIF), an A/D converter, a D/A converter, an interrupt controller (INTC), I/O ports, and I^2C bus interface 3 (IIC3).

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

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

The features of this LSI are listed in table 1.1.

Table 1.1 SH7211 Features

Items	Specification
CPU	• Renesas Technology 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
Operating modes	• Operating modes Extended ROM enabled mode Single-chip mode • Processing states Program execution state Exception handling state Bus mastership release state • Power-down modes Sleep mode Software standby mode Module standby mode

Items	Specification
ROM cache	• Instruction/data separation system • Instruction prefetch cache: Full/set associative • Instruction prefetch miss cache: Full/set associative • Data cache: Full/set associative • Line size: 16 bytes • Hardware prefetch function (continuous/branch prefetch)
Interrupt controller (INTC)	• Nine external interrupt pins (NMI and IRQ7 to IRQ0) • On-chip peripheral interrupts: Priority level set for each module • 16 priority levels available • Register bank enabling fast register saving and restoring in interrupt processing
Bus state controller (BSC)	• Address space divided into eight areas (0 to 7), each a maximum of 64 Mbytes • 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) — Specifying the memory to be connected to each area enables direct connection to SRAM, SRAM with byte selection, SDRAM, and burst ROM (clocked synchronous or asynchronous). The address/data multiplexed I/O (MPX) interface is also available. — Outputs a chip select signal ($\overline{CS0}$ to $\overline{CS7}$) according to the target area ($\overline{CS}$ assert or negate timing can be selected by software) • SDRAM refresh Auto refresh or self refresh mode selectable • SDRAM burst access

Items	Specification
Direct memory access controller (DMAC)	• Eight channels; external request available for four 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
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EXTAL or CK) or crystal resonator • Input clock can be multiplied by 16 (max.) by the internal PLL circuit • Five types of clocks generated: — CPU clock: Maximum 160 MHz — Bus clock: Maximum 40 MHz — Peripheral clock: Maximum 40 MHz — Timer clock: Maximum 80 MHz — AD clock: Maximum 40 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, complementary PWM, and reset-synchronized PWM modes • Synchronization of multiple counters • Complementary PWM output mode — Non-overlapping waveforms output for 3-phase inverter control — Automatic dead time setting — 0% to 100% PWM duty value specifiable — A/D conversion delaying function — Interrupt skipping at crest or trough • Reset-synchronized PWM mode Three-phase PWM waveforms in positive and negative phases can be output with a required duty value • Phase counting mode Two-phase encoder pulse counting available
Multi-function timer pulse unit 2S (MTU2S)	• Subset of MTU2, included in channels 3 to 5 • Operating at 80 MHz max.
Port output enable 2 (POE2)	• High-impedance control of high-current pins at a falling edge or low-level input on the $\overline{\text{POE}}$ pin
Compare match timer (CMT)	• Two-channel 16-bit counters • Four types of clock can be selected (Pϕ/8, Pϕ/32, Pϕ/128, and Pϕ/512) • DMA transfer request or interrupt request can be issued when a compare match occurs
Serial communication interface with FIFO (SCIF)	• Four channels • 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

Items	Specification
I^2C bus interface 3 (IIC3)	• One channel • Master mode and slave mode supported
I/O ports	• Input or output can be selected for each bit
A/D converter (ADC)	• 12-bit resolution • Eight input channels • Sampling can be carried out simultaneously on three channels. • A/D conversion request by the external trigger or timer trigger
D/A converter (DAC)	• 8-bit resolution • Two output channels
User break controller (UBC)	• Four break channels • Addresses, 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 • Realtime branch trace
Advanced user debugger II (AUD- II)	• 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
WAVE interface (WAVEIF)	• Myway Labs realtime CPU scope "WAVE™" (WAVE1.0 Level C) supported
On-chip ROM	• 512 Kbytes
On-chip RAM	• Four pages • 32 Kbytes
Power supply voltage	• Vcc: 1.4 to 1.6 V • VccQ: 3.0 to 3.6 V • AVcc: 4.5 to 5.5 V
Packages	• LQFP2020-144 (0.5 pitch): R5F72115D160FPV

1.2 Block Diagram

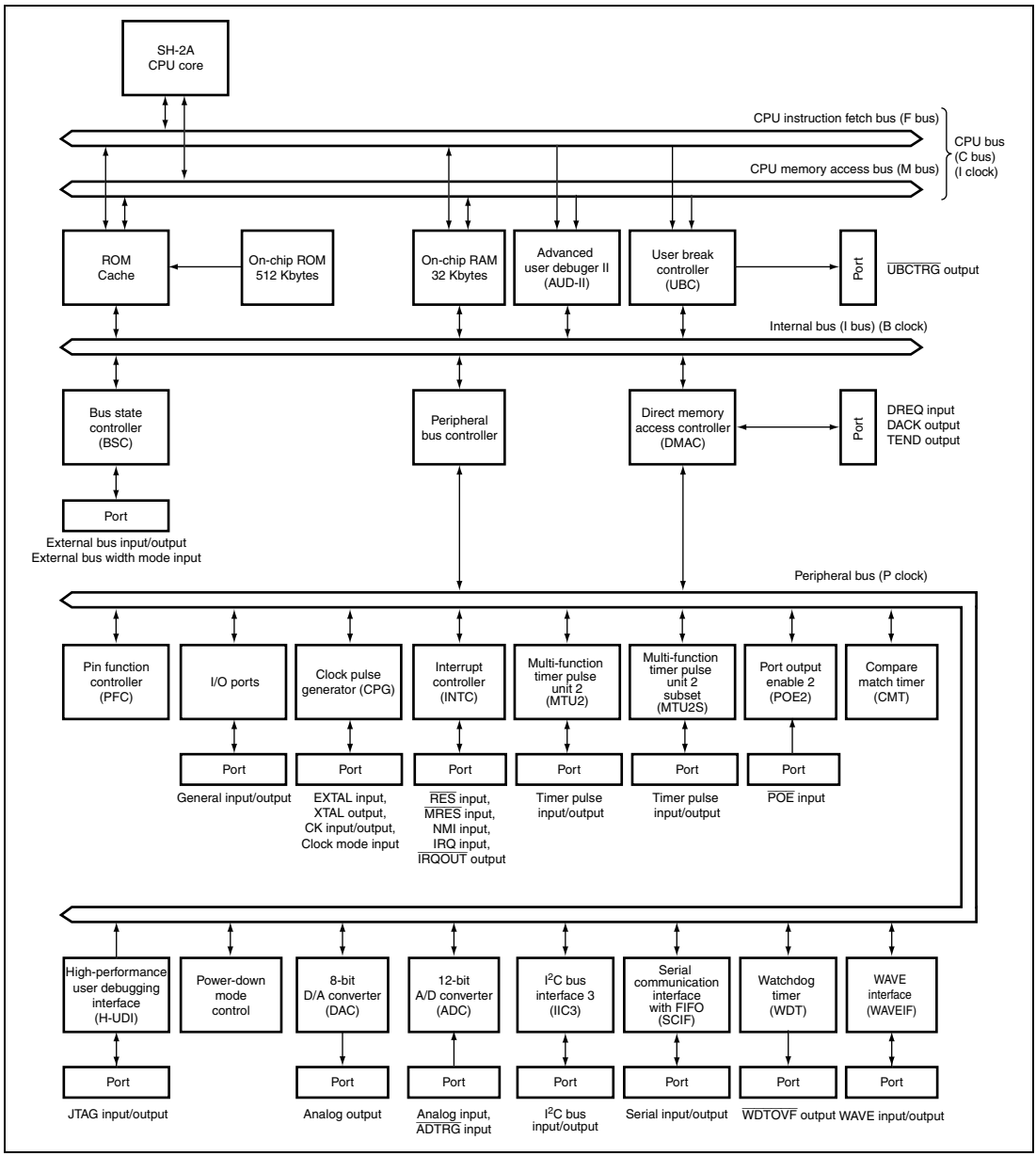
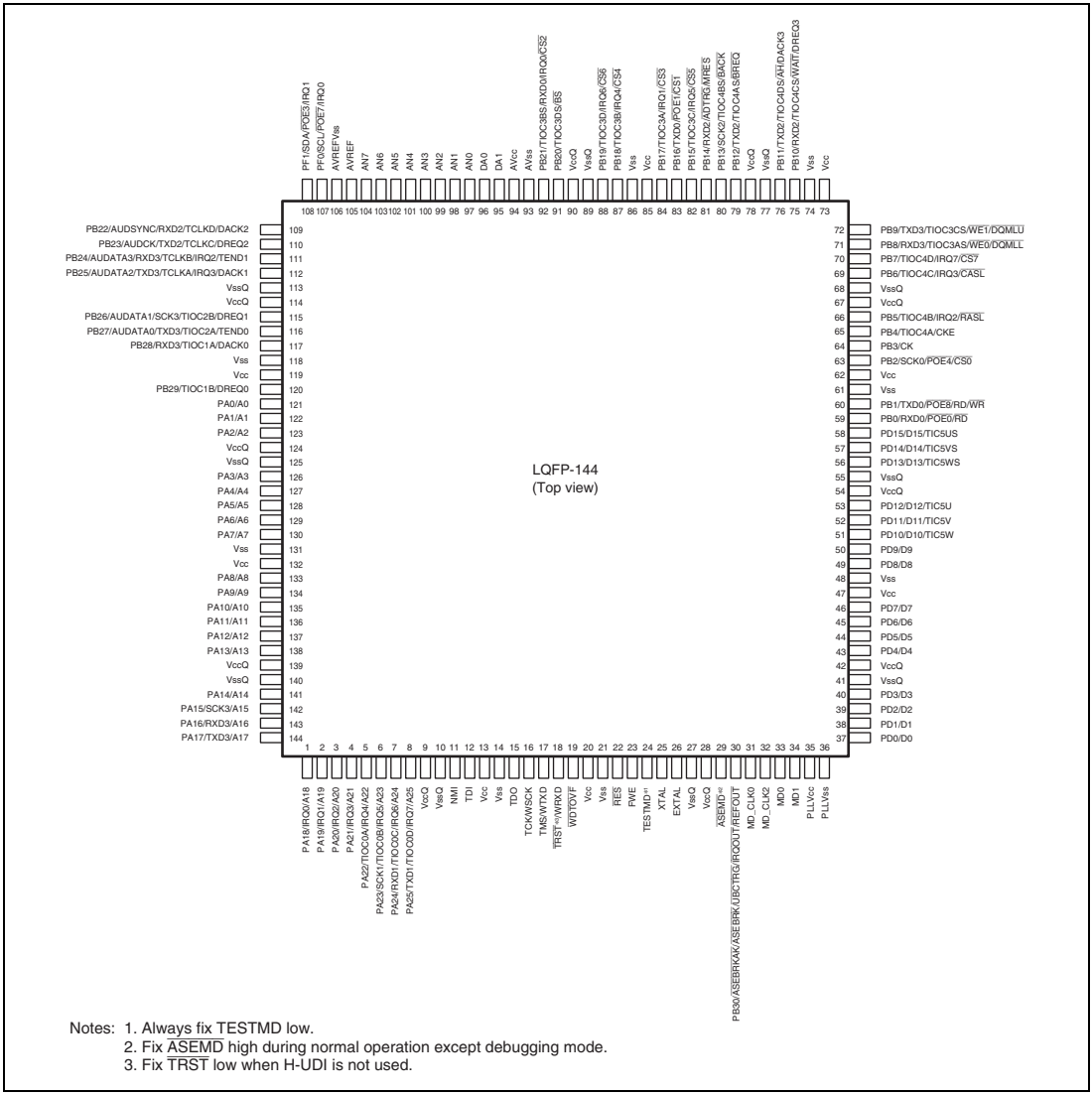


Figure 1.1 Block Diagram

1.3 Pin Arrangement



Notes: 1. Always fix TESTMD low.
2. Fix $\overline{\text{ASEMD}}$ high during normal operation except debugging mode.
3. Fix $\overline{\text{TRST}}$ low when H-UDI is not used.

Figure 1.2 Pin Arrangement

1.4 Pin Functions

Table 1.2 lists functions of each pin.

Table 1.2 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	Vcc	I	Power supply	Power supply pins. All the Vcc pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	Vss	I	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.
	VccQ	I	Power supply for I/O circuits	Power supply for I/O pins. All the VccQ pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	VssQ	I	Ground for I/O circuits	Ground pins for I/O pins. All the VssQ pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.
	PLLVcc	I	Power supply for PLL	Power supply for the on-chip PLL oscillator.
	PLLVss	I	Ground for PLL	Ground pin for the on-chip PLL oscillator.
Clock	EXTAL	I	External clock	Connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	O	Crystal	Connected to a crystal resonator.
	CK	O	System clock	Supplies the system clock to external devices.

Classification	Symbol	I/O	Name	Function
Operating mode control	MD1, MD0	I	Mode set	Sets the operating mode. Do not change the signal levels on these pins during operation.
	MD_CLK2, MD_CLK0	I	Clock mode set	Sets the clock operating mode. Do not change the signal levels on these pins during operation.
	FWE	I	Flash memory write enable	Pin for flash memory. Flash memory can be protected against writing or erasure through this pin.
	$\overline{\text{ASEMD}}$	I	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.
	TESTMD	I	Test mode	Always fix this input pin low. Do not input a high level to this pin. This may cause malfunction or permanent failure of this LSI.
System control	$\overline{\text{RES}}$	I	Power-on reset	This LSI enters the power-on reset state when this signal goes low.
	$\overline{\text{MRES}}$	I	Manual reset	This LSI enters the manual reset state when this signal goes low.
	$\overline{\text{WDTOVF}}$	O	Watchdog timer overflow	Outputs an overflow signal from the WDT.
	$\overline{\text{BREQ}}$	I	Bus-mastership request	A low level is input to this pin when an external device requests the release of the bus mastership.
	$\overline{\text{BACK}}$	O	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	I	Non-maskable interrupt	Non-maskable interrupt request pin. Fix it high when not in use.
	IRQ7 to IRQ0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Level-input or edge-input detection can be selected. When the edge-input detection is selected, the rising edge, falling edge, or both edges can also be selected.
	$\overline{\text{IRQOUT}}$	O	Interrupt request output	Indicates that an interrupt has occurred, enabling external devices to be informed of an interrupt occurrence even while the bus mastership is released.
Address bus	A25 to A0	O	Address bus	Outputs addresses.
Data bus	D15 to D0	I/O	Data bus	Bidirectional data bus.
Bus control	$\overline{\text{CS7}}$ to $\overline{\text{CS0}}$	O	Chip select 7 to 0	Chip-select signals for external memory or devices.
	$\overline{\text{RD}}$	O	Read	Indicates that data is read from an external device.
	RD/$\overline{\text{WR}}$	O	Read/write	Read/write signal.
	$\overline{\text{BS}}$	O	Bus start	Bus-cycle start signal.
	$\overline{\text{AH}}$	O	Address hold	Address hold timing signal for the device that uses the address/data-multiplexed bus.
	$\overline{\text{WAIT}}$	I	Wait	Input signal for inserting a wait cycle into the bus cycles during access to the external space.
	$\overline{\text{WE0}}$	O	Byte select	Indicates a write access to bits 7 to 0 of data of external memory or device.
	$\overline{\text{WE1}}$	O	Byte select	Indicates a write access to bits 15 to 8 of data of external memory or device.

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{\text{DQMLL}}$	O	Byte select	Selects bits D7 to D0 when SDRAM is connected.
	$\overline{\text{DQMLU}}$	O	Byte select	Selects bits D15 to D8 when SDRAM is connected.
	$\overline{\text{RASL}}$	O	RAS	Connected to the $\overline{\text{RAS}}$ pin when SDRAM is connected.
	$\overline{\text{CASL}}$	O	CAS	Connected to the $\overline{\text{CAS}}$ pin when SDRAM is connected.
	CKE	O	CK enable	Connected to the CKE pin when SDRAM is connected.
	$\overline{\text{REFOUT}}$	O	Refresh request	Request signal for refresh execution.
Direct memory access controller (DMAC)	DREQ3 to DREQ0	I	DMA-transfer request	Input pins to receive external requests for DMA transfer.
	DACK3 to DACK0	O	DMA-transfer request accept	Output pins for signals indicating acceptance of external requests from external devices.
	TEND1, TEND0	O	DMA-transfer end output	Output pins for DMA transfer end.
Multi-function timer pulse unit 2 (MTU2)	TCLKA, TCLKB, TCLKC, TCLKD	I	MTU2 timer clock input	External clock input pins for the timer.
	TIOC0A, TIOC0B, TIOC0C, TIOC0D	I/O	MTU2 input capture/output compare (channel 0)	The TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
	TIOC1A, TIOC1B	I/O	MTU2 input capture/output compare (channel 1)	The TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOC2A, TIOC2B	I/O	MTU2 input capture/output compare (channel 2)	The TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.
	TIOC3A, TIOC3B, TIOC3C, TIOC3D	I/O	MTU2 input capture/output compare (channel 3)	The TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.

Classification	Symbol	I/O	Name	Function
Multi-function timer pulse unit 2 (MTU2)	TIOC4A, TIOC4B, TIOC4C, TIOC4D	I/O	MTU2 input capture/output compare (channel 4)	The TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.
	TIC5U, TIC5V, TIC5W	I	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)	$\overline{\text{POE8}}$, $\overline{\text{POE3}}$, $\overline{\text{POE1}}$, $\overline{\text{POE0}}$	I	Port output control	Request signal input to place the MTU2 waveform output pin in the high impedance state.
	$\overline{\text{POE7}}$, $\overline{\text{POE4}}$	I	Port output control	Request signal input to place the MTU2S waveform output pin in the high impedance state.
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 TGRB_4S input capture input/output compare output/PWM output pins.
	TIOC5US, TIOC5VS, TIOC5WS	I	MTU2S input capture (channel 5)	The TGRU_5S, TGRV_5S, and TGRW_5S input capture input/dead time compensation input pins.
	$\overline{\text{RTS3}}$	O	Transmit request	Modem control pin.
	$\overline{\text{CTS3}}$	I	Transmit enable	Modem control pin.
I^2C bus interface 3 (IIC3)	SCL	I/O	Serial clock pin	Serial clock input/output pin.
	SDA	I/O	Serial data pin	Serial data input/output pin.
A/D converter (ADC)	AN7 to AN0	I	Analog input pins	Analog input pins.
	$\overline{\text{ADTRG}}$	I	A/D conversion trigger input	External trigger input pin for starting A/D conversion.
	AVcc	I	Analog power supply	Power supply pin for the A/D converter. Connect this pin to the system power supply (VccQ) when the A/D converter is not used.
	AVREF	I	Analog reference power supply	Reference voltage pin for the A/D converter. Connect this pin to the system power supply (VccQ) when the A/D converter is not used.

Classification	Symbol	I/O	Name	Function
A/D converter (ADC)	AVss	I	Analog ground	Ground pin for the A/D converter. Connect this pin to the system power supply (VssQ) when the A/D converter is not used.
	AVREFVss	I	Analog reference ground	Reference ground pin for the A/D converter. Connect this pin to the system power supply (VssQ) when the A/D converter is not used.
D/A converter (DAC)	DA1, DA0	O	Analog output pins	Analog output pins.
I/O ports	PA25 to PA0	I/O	General port	26-bit general input/output port pins.
	PB30 to PB0	I/O	General port	31-bit general input/output port pins.
	PD15 to PD0	I/O	General port	16-bit general input/output port pins.
	PF1, PF0	I	General port	2-bit general input/output port pins.
User debugging interface (H-UDI)	TCK	I	Test clock	Test-clock input pin.
	TMS	I	Test mode select	Test-mode select signal input pin.
	TDI	I	Test data input	Serial input pin for instructions and data.
	TDO	O	Test data output	Serial output pin for instructions and data.
	$\overline{\text{TRST}}$	I	Test reset	Initialization-signal input pin. Input a low level when not using the H-UDI.
Advanced user debugger (AUD)	AUDATA3 to AUDATA0	I/O	AUD data	Branch destination/source address output pin
	AUDCK	I/O	AUD clock	Sync clock output pin
	AUDSYNC	I/O	AUD sync signal	Data start-position acknowledge-signal output pin
Emulator interface	$\overline{\text{ASEBRKAK}}$	O	Break mode acknowledge	Indicates that the E10A-USB emulator has entered its break mode.
	$\overline{\text{ASEBRK}}$	I	Break request	E10A-USB emulator break input pin.
User break controller (UBC)	$\overline{\text{UBCTRG}}$	O	User break trigger output	Trigger output pin for UBC condition match.
WAVE interface (WAVEIF)	WSCK	O	Clock output	Interface pin to support Myway Labs realtime CPU scope “WAVE™” (WAVE1.0 Level C)
	WRXD	I	Receive data	
	WTXD	O	Transmit data	

Section 2 CPU

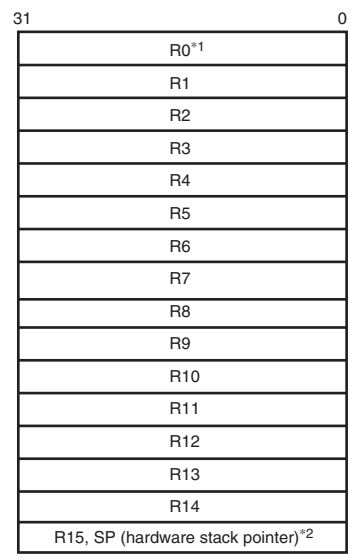
2.1 Register Configuration

The register set consists of sixteen 32-bit general registers, four 32-bit control registers, and four 32-bit system registers.

2.1.1 General Registers

Figure 2.1 shows the general registers.

The sixteen 32-bit general registers are numbered R0 to R15. General registers 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.



Notes: 1. R0 functions as an index register in the indexed register indirect addressing mode and indexed GBR indirect addressing mode. In some instructions, R0 functions as a fixed source register or destination register.
2. R15 functions as a hardware stack pointer (SP) during exception processing.

Figure 2.1 General Registers

2.1.2 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.

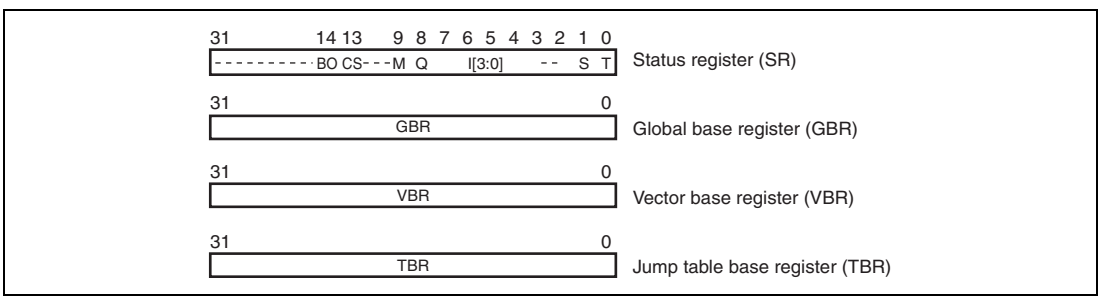


Figure 2.2 Control Registers

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

(2) Global Base Register (GBR)

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

(3) Vector Base Register (VBR)

VBR is referenced as the branch destination base address in the event of an exception or an interrupt.

(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.

2.1.3 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.

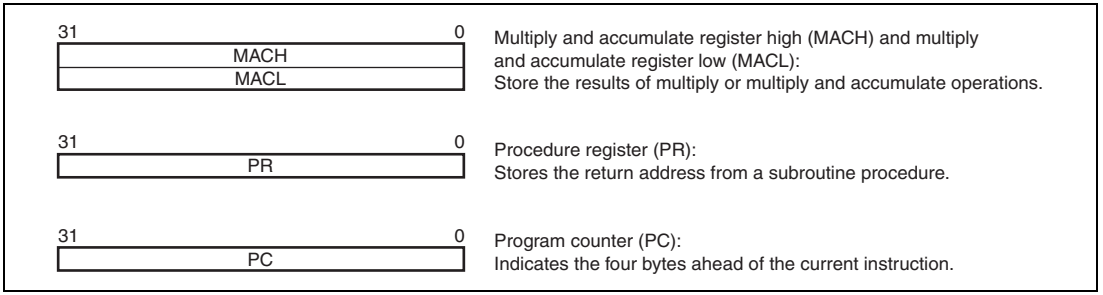


Figure 2.3 System Registers

(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 of the instruction being executed.

2.1.4 Register Banks

For the nineteen 32-bit registers comprising general registers R0 to R14, control register GBR, and system registers MACH, MACL, and PR, high-speed register saving and restoration can be carried out using a register bank. The register contents are automatically saved in the bank after the CPU accepts an interrupt that uses a register bank. Restoration from the bank is executed by issuing a RESBANK instruction in an interrupt processing routine.

For details, refer to the SH-2A, SH2A-FPU Software Manual.

2.1.5 Initial Values of Registers

Table 2.1 lists the values of the registers after a reset.

Table 2.1 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 other bits 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

2.2 Data Formats

2.2.1 Data Format in Registers

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

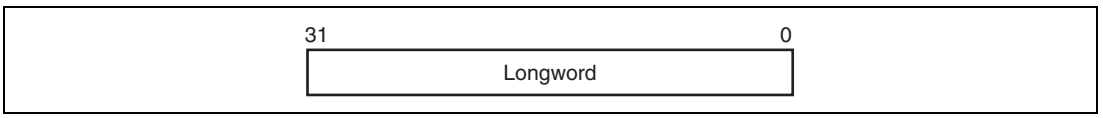


Figure 2.4 Data Format in Registers

2.2.2 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 2.5.

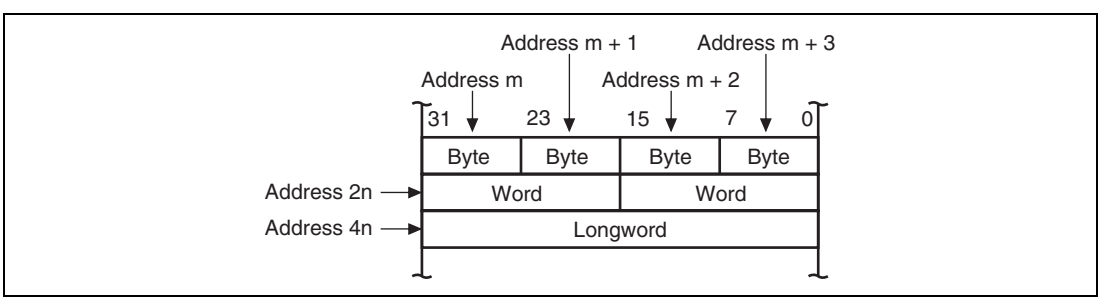


Figure 2.5 Data Formats in Memory

2.2.3 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 2.3.1 (10), Immediate Data.

2.3 Instruction Features

2.3.1 RISC-Type Instruction Set

Instructions are RISC type. This section details their 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 additionally features 32-bit fixed-length instructions, improving performance and ease of use.

(3) One Instruction per State

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

(4) Data Length

Longword is the standard data length for all operations. 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 2.2 Sign Extension of Word Data

SH2-A CPU		Description	Example of Other CPU
MOV.W	@(disp,PC),R1	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
ADD	R1,R0		
	.....		
.DATA.W	H'1234		

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 2.3 Delayed Branch Instructions

SH-2A 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 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 2.4 T Bit

SH-2A 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, 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 2.5 Immediate Data Accessing

Classification	SH-2A 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, 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 2.6 Absolute Address Accessing

Classification	SH-2A 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.

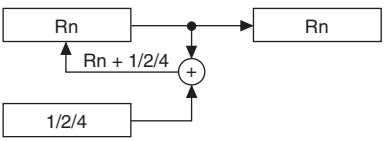
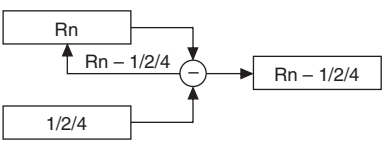
Table 2.7 Displacement Accessing

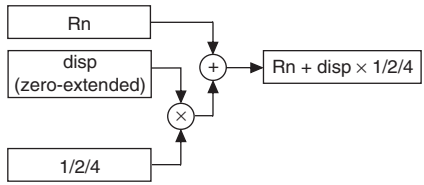
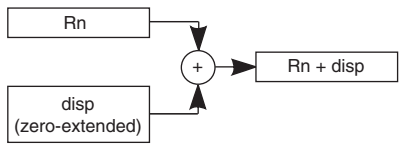
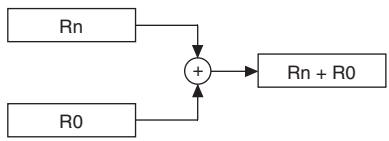
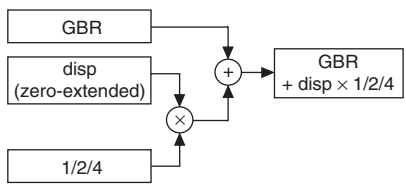
Classification	SH-2A CPU		Example of Other CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

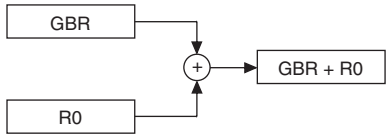
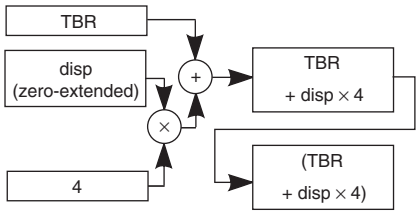
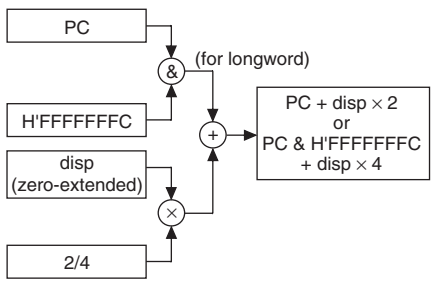
2.3.2 Addressing Modes

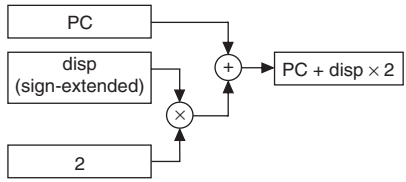
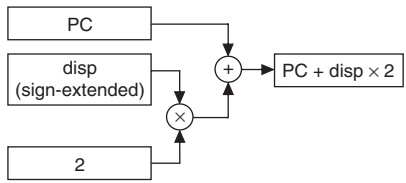
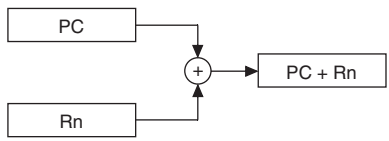
Addressing modes and effective address calculation are as follows:

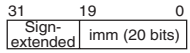
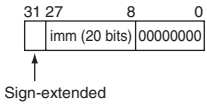
Table 2.8 Addressing Modes and Effective Addresses

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
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)

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
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
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

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
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)
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

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
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
	`Rn`	The effective address is the sum of PC value and Rn. 	PC + Rn

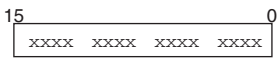
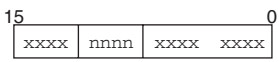
Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Immediate	#imm:20	The 20-bit immediate data (imm) for the MOVI20 instruction is sign-extended. 	—
		The 20-bit immediate data (imm) for the MOVI20S instruction is shifted by eight bits to the left, the upper bits are sign-extended, 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.	—

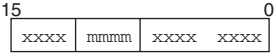
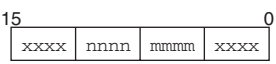
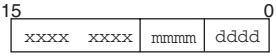
2.3.3 Instruction Format

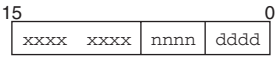
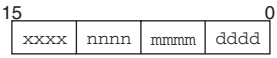
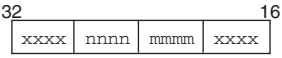
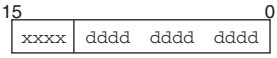
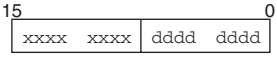
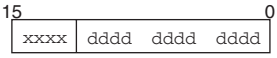
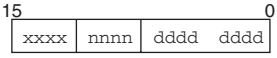
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

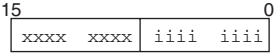
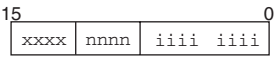
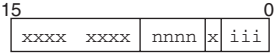
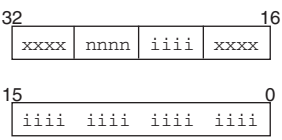
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

Table 2.9 Instruction Formats

Instruction Formats	Source Operand	Destination Operand	Example
0 format 	—	—	`NOP`
n format 	—	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 	dddddddddddd: 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 	iiiiiiiiiiiiiiiiii: Immediate	nnnn: Register direct	MOVI20 #imm20, Rn
nid format 	nnnnddddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnnddddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12,Rn)

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

2.4 Instruction Set

2.4.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their classification.

Table 2.10 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	
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	

Classification	Types	Operation Code	Function	No. of Instructions
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	
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:	91			197

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 States	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		Explanation of Symbols —: 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.

2.4.2 Data Transfer Instructions

Table 2.11 Data Transfer Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
MOV	#imm,Rn	1110nnnniiiiiiii	imm → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(disp,PC),Rn	1001nnnndddddddd	(disp × 2 + PC) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(disp,PC),Rn	1101nnnndddddddd	(disp × 4 + PC) → Rn	1	—	Yes	Yes	Yes
MOV	Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm,Rn	0110nnnnmmmm0000	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@Rm,Rn	0110nnnnmmmm0001	(Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.W	Rm,@-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—	Yes	Yes	Yes
MOV.B	@Rm+,Rn	0110nnnnmmmm0100	(Rm) → sign extension → Rn, Rm + 1 → Rm	1	—	Yes	Yes	Yes
MOV.W	@Rm+,Rn	0110nnnnmmmm0101	(Rm) → sign extension → Rn, Rm + 2 → Rm	1	—	Yes	Yes	Yes
MOV.L	@Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,Rn)	10000000nnnndddd	R0 → (disp + Rn)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,Rn)	10000001nnnndddd	R0 → (disp × 2 + Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@(disp,Rn)	0001nnnnmmmmdddd	Rm → (disp × 4 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(disp,Rm),R0	10000100mmmmdddd	(disp + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,Rm),R0	10000101mmmmdddd	(disp × 2 + Rm) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,Rm),Rn	0101nnnnmmmmdddd	(disp × 4 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.L	Rm,@(R0,Rn)	0000nnnnmmmm0110	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes
MOV.B	@(R0,Rm),Rn	0000nnnnmmmm1100	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.W	@(R0,Rm),Rn	0000nnnnmmmm1101	(R0 + Rm) → sign extension → Rn	1	—	Yes	Yes	Yes
MOV.L	@(R0,Rm),Rn	0000nnnnmmmm1110	(R0 + Rm) → Rn	1	—	Yes	Yes	Yes
MOV.B	R0,@(disp,GBR)	11000000dddddddd	R0 → (disp + GBR)	1	—	Yes	Yes	Yes
MOV.W	R0,@(disp,GBR)	11000001dddddddd	R0 → (disp × 2 + GBR)	1	—	Yes	Yes	Yes
MOV.L	R0,@(disp,GBR)	11000010dddddddd	R0 → (disp × 4 + GBR)	1	—	Yes	Yes	Yes
MOV.B	@(disp,GBR),R0	11000100dddddddd	(disp + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.W	@(disp,GBR),R0	11000101dddddddd	(disp × 2 + GBR) → sign extension → R0	1	—	Yes	Yes	Yes
MOV.L	@(disp,GBR),R0	11000110dddddddd	(disp × 4 + GBR) → R0	1	—	Yes	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
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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
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	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
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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
MOVRT	Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT	Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	Yes
MOVU.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 1000ddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 1001ddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes
NOTT		0000000001101000	~T → T	1	Operation result			Yes
PREF	@Rn	0000nnnn10000011	(Rn) → operand cache	1	—		Yes	Yes
SWAP.B	Rm,Rn	0110nnnnmmmm1000	Rm → swap lower 2 bytes → Rn	1	—	Yes	Yes	Yes
SWAP.W	Rm,Rn	0110nnnnmmmm1001	Rm → swap upper and lower words → Rn	1	—	Yes	Yes	Yes
XTRCT	Rm,Rn	0010nnnnmmmm1101	Middle 32 bits of Rm:Rn → Rn	1	—	Yes	Yes	Yes

2.4.3 Arithmetic Operation Instructions

Table 2.12 Arithmetic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
ADD	Rm,Rn	0011nnnnmmmm1100	Rn + Rm → Rn	1	—	Yes	Yes	Yes
ADD	#imm,Rn	0111nnnniiiiiiii	Rn + imm → Rn	1	—	Yes	Yes	Yes
ADDC	Rm,Rn	0011nnnnmmmm1110	Rn + Rm + T → Rn, carry → T	1	Carry	Yes	Yes	Yes
ADDV	Rm,Rn	0011nnnnmmmm1111	Rn + Rm → Rn, overflow → T	1	Over-flow	Yes	Yes	Yes
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Com-parison result	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
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	Yes
DIV0S	Rm,Rn	0010nnnnmmmm0111	MSB of Rn → Q, MSB of Rm → M, M ^ Q → T	1	Calculation result	Yes	Yes	Yes
DIV0U		0000000000011001	0 → M/Q/T	1	0	Yes	Yes	Yes
DIVS	R0,Rn	0100nnnn10010100	Signed operation of Rn ÷ R0 → Rn 32 ÷ 32 → 32 bits	36	—			Yes
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	Yes
DMULU.L	Rm,Rn	0011nnnnmmmm0101	Unsigned operation of Rn × Rm → MACH, MACL 32 × 32 → 64 bits	2	—	Yes	Yes	Yes
DT	Rn	0100nnnn00010000	Rn – 1 → Rn When Rn is 0, 1 → T When Rn is not 0, 0 → T	1	Comparison result	Yes	Yes	Yes
EXTS.B	Rm,Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes
EXTS.W	Rm,Rn	0110nnnnmmmm1111	Word in Rm is sign-extended → Rn	1	—	Yes	Yes	Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
EXTU.W	Rm,Rn	0110nnnnmmmm1101	Word in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes
MAC.L	@Rm+,@Rn+	0000nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 32 × 32 + 64 → 64 bits	4	—	Yes	Yes	Yes
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	Signed operation of (Rn) × (Rm) + MAC → MAC 16 × 16 + 64 → 64 bits	3	—	Yes	Yes	Yes
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32 × 32 → 32 bits	2	—	Yes	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	Yes
MULU.W	Rm,Rn	0010nnnnmmmm1110	Unsigned operation of Rn × Rm → MACL 16 × 16 → 32 bits	1	—	Yes	Yes	Yes
NEG	Rm,Rn	0110nnnnmmmm1011	0-Rm → Rn	1	—	Yes	Yes	Yes
NEGC	Rm,Rn	0110nnnnmmmm1010	0-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUB	Rm,Rn	0011nnnnmmmm1000	Rn-Rm → Rn	1	—	Yes	Yes	Yes
SUBC	Rm,Rn	0011nnnnmmmm1010	Rn-Rm-T → Rn, borrow → T	1	Borrow	Yes	Yes	Yes
SUBV	Rm,Rn	0011nnnnmmmm1011	Rn-Rm → Rn, underflow → T	1	Over-flow	Yes	Yes	Yes

2.4.4 Logic Operation Instructions

Table 2.13 Logic Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
AND	Rm,Rn	0010nnnnmmmm1001	Rn & Rm → Rn	1	—	Yes	Yes	Yes
AND	#imm,R0	11001001iiiiiiii	R0 & imm → R0	1	—	Yes	Yes	Yes
AND.B	#imm,@(R0,GBR)	11001101iiiiiiii	(R0 + GBR) & imm → (R0 + GBR)	3	—	Yes	Yes	Yes
NOT	Rm,Rn	0110nnnnmmmm0111	~Rm → Rn	1	—	Yes	Yes	Yes
OR	Rm,Rn	0010nnnnmmmm1011	Rn | Rm → Rn	1	—	Yes	Yes	Yes
OR	#imm,R0	11001011iiiiiiii	R0 | imm → R0	1	—	Yes	Yes	Yes
OR.B	#imm,@(R0,GBR)	11001111iiiiiiii	(R0 + GBR) | imm → (R0 + GBR)	3	—	Yes	Yes	Yes
TAS.B	@Rn	0100nnnn00011011	When (Rn) is 0, 1 → T Otherwise, 0 → T, 1 → MSB of(Rn)	3	Test result	Yes	Yes	Yes
TST	Rm,Rn	0010nnnnmmmm1000	Rn & Rm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	Yes	Yes
TST	#imm,R0	11001000iiiiiiii	R0 & imm When the result is 0, 1 → T Otherwise, 0 → T	1	Test result	Yes	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	Yes
XOR	Rm,Rn	0010nnnnmmmm1010	Rn ^ Rm → Rn	1	—	Yes	Yes	Yes
XOR	#imm,R0	11001010iiiiiiii	R0 ^ imm → R0	1	—	Yes	Yes	Yes
XOR.B	#imm,@(R0,GBR)	11001110iiiiiiii	(R0 + GBR) ^ imm → (R0 + GBR)	3	—	Yes	Yes	Yes

2.4.5 Shift Instructions

Table 2.14 Shift Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
ROTL	Rn	0100nnnn00000100	T ← Rn ← MSB	1	MSB	Yes	Yes	Yes
ROTR	Rn	0100nnnn00000101	LSB → Rn → T	1	LSB	Yes	Yes	Yes
ROTCL	Rn	0100nnnn00100100	T ← Rn ← T	1	MSB	Yes	Yes	Yes
ROTCR	Rn	0100nnnn00100101	T → Rn → T	1	LSB	Yes	Yes	Yes
SHAD	Rm,Rn	0100nnnnmmmm1100	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [MSB → Rn]	1	—		Yes	Yes
SHAL	Rn	0100nnnn00100000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHAR	Rn	0100nnnn00100001	MSB → Rn → T	1	LSB	Yes	Yes	Yes
SHLD	Rm,Rn	0100nnnnmmmm1101	When Rm ≥ 0, Rn << Rm → Rn When Rm < 0, Rn >> |Rm| → [0 → Rn]	1	—		Yes	Yes
SHLL	Rn	0100nnnn00000000	T ← Rn ← 0	1	MSB	Yes	Yes	Yes
SHLR	Rn	0100nnnn00000001	0 → Rn → T	1	LSB	Yes	Yes	Yes
SHLL2	Rn	0100nnnn00001000	Rn << 2 → Rn	1	—	Yes	Yes	Yes
SHLR2	Rn	0100nnnn00001001	Rn >> 2 → Rn	1	—	Yes	Yes	Yes
SHLL8	Rn	0100nnnn00011000	Rn << 8 → Rn	1	—	Yes	Yes	Yes
SHLR8	Rn	0100nnnn00011001	Rn >> 8 → Rn	1	—	Yes	Yes	Yes
SHLL16	Rn	0100nnnn00101000	Rn << 16 → Rn	1	—	Yes	Yes	Yes
SHLR16	Rn	0100nnnn00101001	Rn >> 16 → Rn	1	—	Yes	Yes	Yes

2.4.6 Branch Instructions

Table 2.15 Branch Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
BF	label	10001011dddddddd	When T = 0, disp × 2 + PC → PC, When T = 1, nop	3/1*	—	Yes	Yes	Yes
BF/S	label	10001111dddddddd	Delayed branch When T = 0, disp × 2 + PC → PC, When T = 1, nop	2/1*	—	Yes	Yes	Yes
BT	label	10001001dddddddd	When T = 1, disp × 2 + PC → PC, When T = 0, nop	3/1*	—	Yes	Yes	Yes
BT/S	label	10001101dddddddd	Delayed branch When T = 1, disp × 2 + PC → PC, When T = 0, nop	2/1*	—	Yes	Yes	Yes
BRA	label	1010dddddddddddd	Delayed branch, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BRAF	Rm	0000mmmm00100011	Delayed branch, Rm + PC → PC	2	—	Yes	Yes	Yes
BSR	label	1011dddddddddddd	Delayed branch, PC → PR, disp × 2 + PC → PC	2	—	Yes	Yes	Yes
BSRF	Rm	0000mmmm00000011	Delayed branch, PC → PR, Rm + PC → PC	2	—	Yes	Yes	Yes
JMP	@Rm	0100mmmm00101011	Delayed branch, Rm → PC	2	—	Yes	Yes	Yes
JSR	@Rm	0100mmmm00001011	Delayed branch, PC → PR, Rm → PC	2	—	Yes	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	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.

2.4.7 System Control Instructions

Table 2.16 System Control Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
CLRT		0000000000001000	0 → T	1	0	Yes	Yes	Yes
CLRMAC		0000000000101000	0 → MACH,MACL	1	—	Yes	Yes	Yes
LDBANK	@Rm,R0	0100mmmm11100101	(Specified register bank entry) → R0	6	—			Yes
LDC	Rm,SR	0100mmmm00001110	Rm → SR	3	LSB	Yes	Yes	Yes
LDC	Rm,TBR	0100mmmm01001010	Rm → TBR	1	—			Yes
LDC	Rm,GBR	0100mmmm00011110	Rm → GBR	1	—	Yes	Yes	Yes
LDC	Rm,VBR	0100mmmm00101110	Rm → VBR	1	—	Yes	Yes	Yes
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	5	LSB	Yes	Yes	Yes
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS	Rm,MACH	0100mmmm00001010	Rm → MACH	1	—	Yes	Yes	Yes
LDS	Rm,MACL	0100mmmm00011010	Rm → MACL	1	—	Yes	Yes	Yes
LDS	Rm,PR	0100mmmm00101010	Rm → PR	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—	Yes	Yes	Yes
LDS.L	@Rm+,PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—	Yes	Yes	Yes
NOP		0000000000001001	No operation	1	—	Yes	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	Yes
SETT		0000000000011000	1 → T	1	1	Yes	Yes	Yes
SLEEP		0000000000011011	Sleep	5	—	Yes	Yes	Yes
STBANK	R0,@Rn	0100nnnn11100001	R0 → (specified register bank entry)	7	—			Yes
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Yes	Yes
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
STC	GBR,Rn	0000nnnn00010010	GBR → Rn	1	—	Yes	Yes	Yes
STC	VBR,Rn	0000nnnn00100010	VBR → Rn	1	—	Yes	Yes	Yes
STC.L	SR,@-Rn	0100nnnn00000011	Rn-4 → Rn, SR → (Rn)	2	—	Yes	Yes	Yes
STC.L	GBR,@-Rn	0100nnnn00010011	Rn-4 → Rn, GBR → (Rn)	1	—	Yes	Yes	Yes
STC.L	VBR,@-Rn	0100nnnn00100011	Rn-4 → Rn, VBR → (Rn)	1	—	Yes	Yes	Yes
STS	MACH,Rn	0000nnnn00001010	MACH → Rn	1	—	Yes	Yes	Yes
STS	MACL,Rn	0000nnnn00011010	MACL → Rn	1	—	Yes	Yes	Yes
STS	PR,Rn	0000nnnn00101010	PR → Rn	1	—	Yes	Yes	Yes
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4 → Rn, MACH → (Rn)	1	—	Yes	Yes	Yes
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4 → Rn, MACL → (Rn)	1	—	Yes	Yes	Yes
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4 → Rn, PR → (Rn)	1	—	Yes	Yes	Yes
TRAPA	#imm	11000011iiiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	5	—	Yes	Yes	Yes

Notes: 1. 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.

2.4.8 Bit Manipulation Instructions

Table 2.17 Bit Manipulation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp + Rn)) & 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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

2.5 Processing States

The CPU has five processing states: reset, exception handling, bus-released, program execution, and power-down. Figure 2.6 shows the transitions between the states.

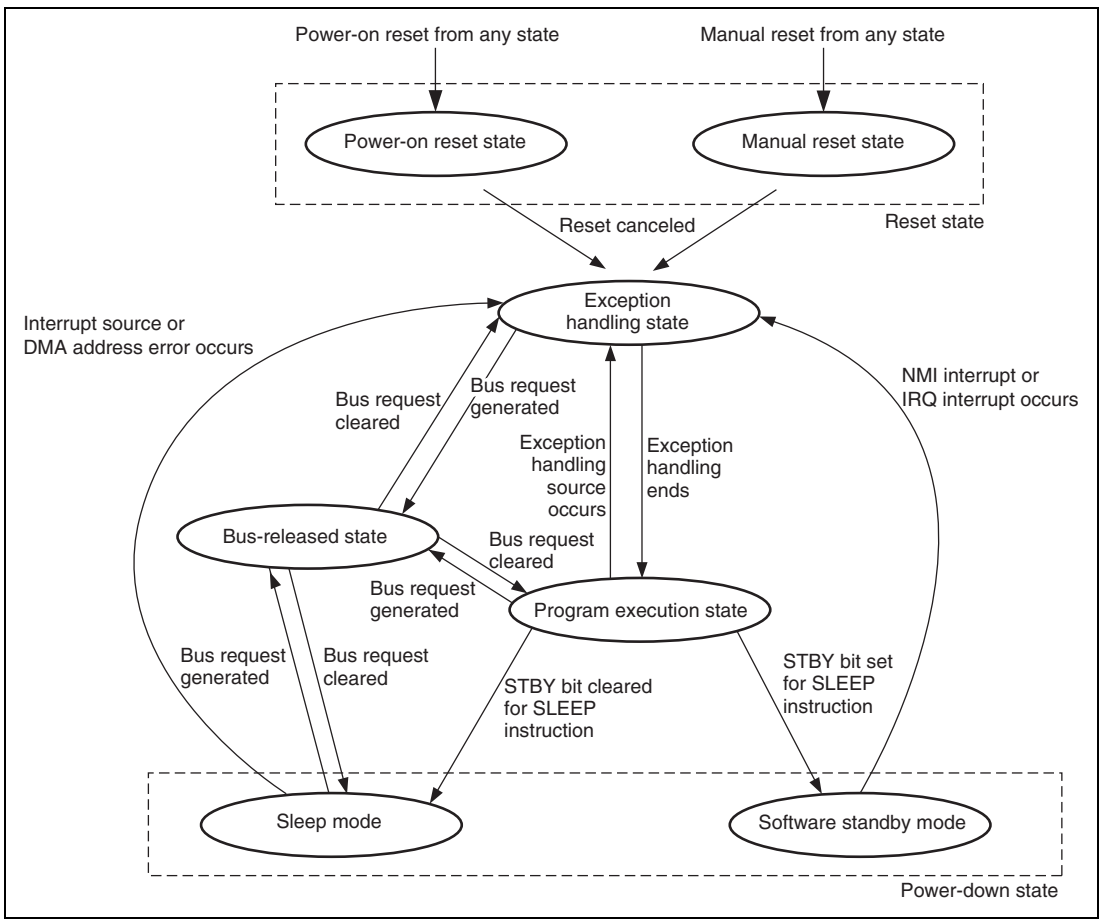


Figure 2.6 Transitions between Processing States

(1) Reset State

In the reset 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 alter 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.

(5) Bus-Released State

In the bus-released state, the CPU releases bus to a device that has requested it.

Section 3 MCU Operating Modes

3.1 Selection of Operating Modes

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

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

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

Table 3.1 Selection of Operating Modes

	Pin Setting					Bus Width of CS0 Space
Mode No.	FWE	MD1	MD0	Mode Name	On-Chip ROM	SH7211F
Mode 0	0	0	0	MCU extension mode 0	Not active	16
Mode 1	0	0	1	MCU extension mode 1	Not active	8
Mode 2	0	1	0	MCU extension mode 2	Active	Set by CS0BCR in BSC
Mode 3	0	1	1	Single chip mode	Active	—
Mode 4*	1	0	0	Boot mode	Active	Set by CS0BCR in BSC
Mode 5*	1	0	1	User boot mode	Active	Set by CS0BCR in BSC
Mode 6*	1	1	0	User program mode	Active	Set by CS0BCR in BSC

Note: * Flash memory programming mode.

3.2 Input/Output Pins

Table 3.2 describes the configuration of operating mode related pin.

Table 3.2 Pin Configuration

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

3.3 Operating Modes

3.3.1 Mode 0 (MCU Extension Mode 0)

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

3.3.2 Mode 1 (MCU Extension Mode 1)

In this mode, CS0 space becomes external memory spaces with 8-bit bus width.

3.3.3 Mode 2 (MCU Extension Mode 2)

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

3.3.4 Mode 3 (Single Chip Mode)

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

3.4 Address Map

The address map for the operating modes is shown in figure 3.1.

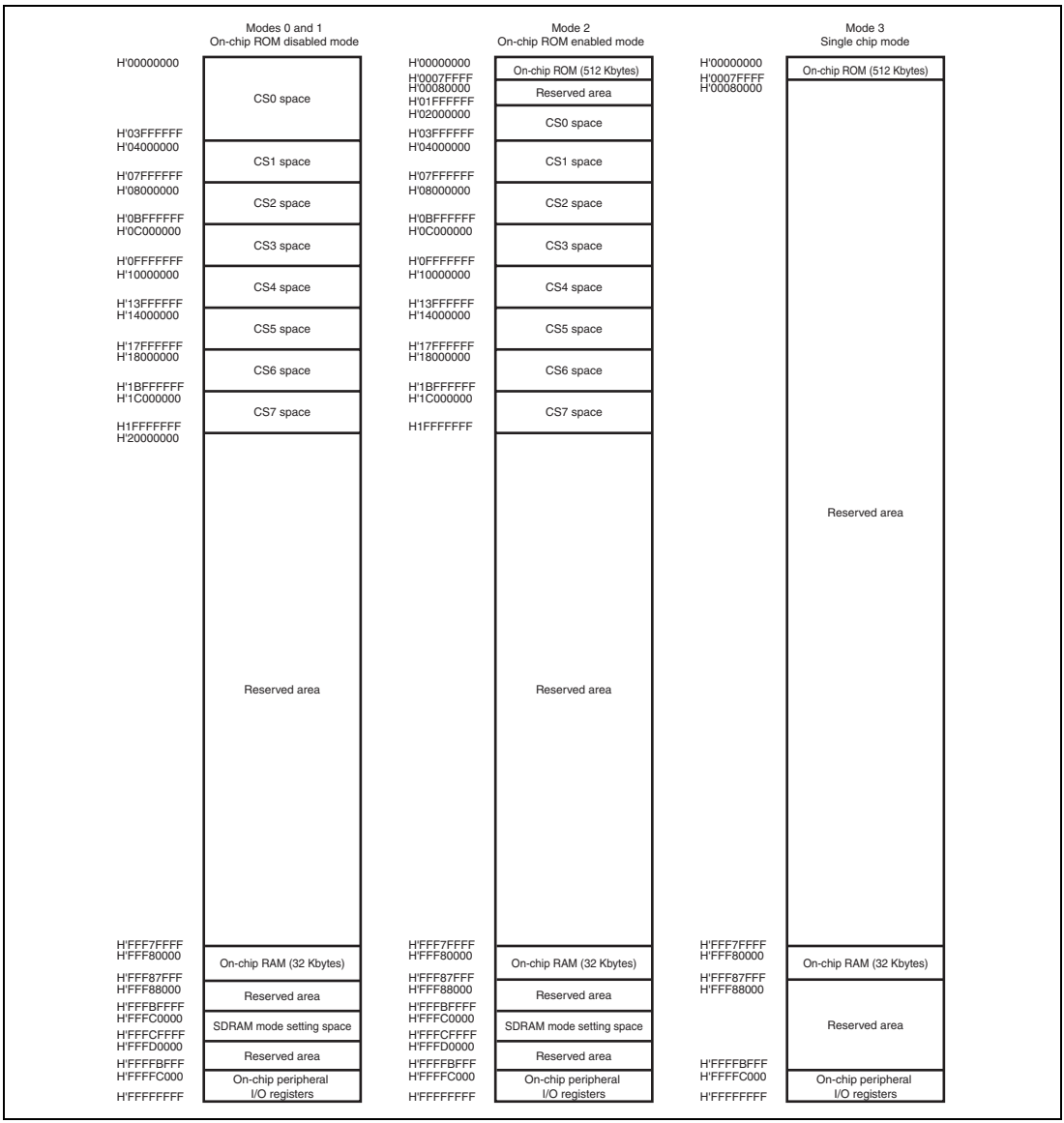


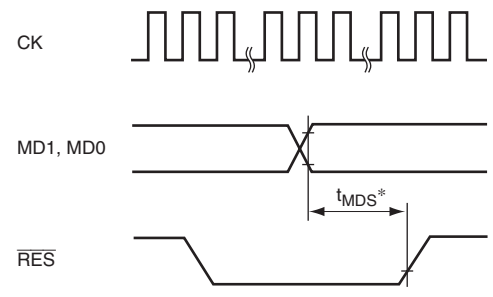
Figure 3.1 Address Map for Each Operating Mode

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 23, Power-Down Modes.

3.6 Note on Changing Operating Mode

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



Note: * See section 27.4.2, Control Signal Timing.

Figure 3.2 Reset Input Timing when Changing Operating Mode

Section 4 Clock Pulse Generator (CPG)

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

4.1 Features

- Clock operating modes

  Either the internal crystal resonator or the input on the external clock-signal line can be selected.
- Five clocks generated independently

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

  Internal and peripheral clock frequencies can be changed independently using the PLL (phase locked loop) circuits and divider circuits 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 23, Power-Down Modes.

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

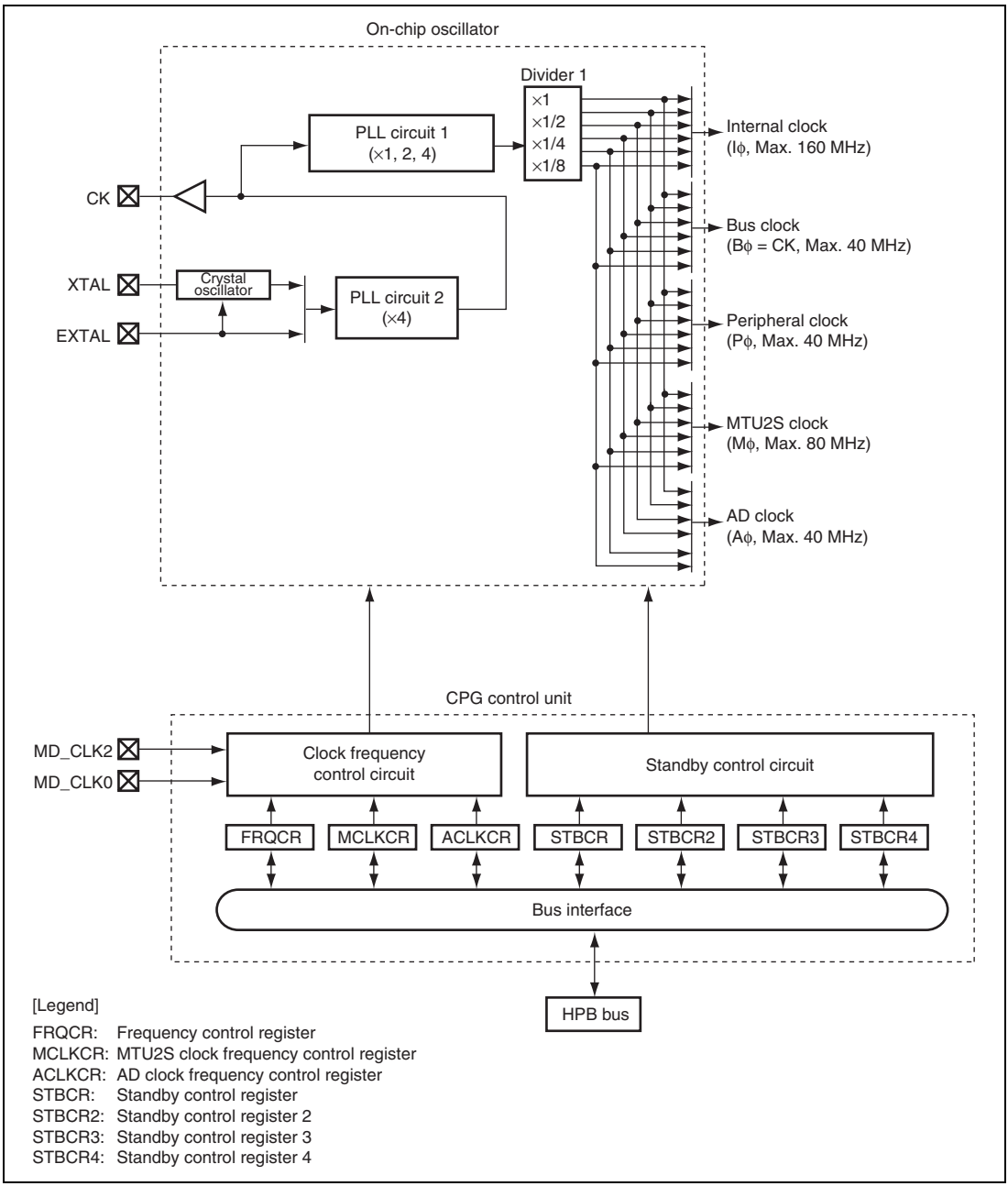


Figure 4.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

(1) PLL Circuit 1

PLL circuit 1 multiplies the input clock frequency from the CK pin by 1, 2, or 4. The multiplication rate is set by the frequency control register. When this is done, the phase of the rising edge of the internal clock (Iϕ) is controlled so that it will agree with the phase of the rising edge of the CK pin.

(2) PLL Circuit 2

PLL circuit 2 multiplies the input clock frequency from the crystal oscillator or EXTAL pin by 4. The multiplication rate is fixed according to the clock operating mode. The clock operating mode is specified by the MD_CLK0 and MD_CLK2 pins. For details on the clock operating mode, see table 4.2.

(3) 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.

(4) Divider 1

Divider 1 generates a clock signal at the operating frequency used by the internal clock (Iϕ), the bus clock (Bϕ), the peripheral clock (Pϕ), the MTU2S clock (Mϕ), or the AD clock (Aϕ). The operating frequency can be 1, 1/2, 1/4, or 1/8 times the output frequency of PLL circuit 1. However, set the internal clock (Iϕ) so that its frequency is not less than the clock frequency of the CK pin, and set the peripheral clock (Pϕ) so that its frequency is not more than the clock frequency of the CK pin. The division ratio is set in the frequency control register (FRQCR).

(5) Clock Frequency Control Circuit

The clock frequency control circuit controls the clock frequency using the MD_CLK0 and MD_CLK2 pins and the frequency control register (FRQCR).

(6) 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.

(7) Frequency Control Register (FRQCR)

The frequency control register (FRQCR) has control bits assigned for the following functions: clock output/non-output from the CK pin during software standby mode, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the internal clock (Iϕ) and the peripheral clock (Pϕ).

(8) MTU2S Clock Frequency Control Register (MCLKCR)

The MTU2S clock frequency control register (MCLKCR) has control bits assigned for the following functions: MTU2S clock (Mϕ) output/non-output and the frequency division ratio.

(9) AD Clock Frequency Control Register (ACLKCR)

The AD clock frequency control register (ACLKCR) has control bits assigned for the following functions: AD clock (Aϕ) output/non-output and the frequency division ratio.

(10) Standby Control Register

The standby control register has bits for controlling the power-down modes. See section 23, Power-Down Modes, for more information.

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 (Clock Operating Mode 6)
Mode control pins	MD_CLK0	Input	Sets the clock operating mode.
	MD_CLK2	Input	Sets the clock operating mode.
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 high impedance.

4.3 Clock Operating Modes

Table 4.2 shows the relationship between the combinations of the mode control pins (MD_CLK2 and MD_CLK0) and the clock operating modes. Table 4.3 shows the usable frequency ranges in the clock operating modes.

Table 4.2 Clock Operating Modes

	Pin Values		Clock I/O				
Mode	MD_CLK2	MD_CLK0	Source	Output	PLL Circuit 2 On/Off	PLL Circuit 1 On/Off	CK Frequency
6	1	0	EXTAL or crystal resonator	CK	On (× 4)	On (× 1, 2, 4)	(EXTAL or crystal resonator) × 4

- Mode 6

  The frequency of the signal received from the EXTAL pin or crystal oscillator is quadrupled by the PLL circuit 2 before it is supplied to the LSI as the clock signal. This allows a crystal with a lower frequency to be used. Either a crystal resonator with a frequency in the range from 8 to 10 MHz or an external signal in the same frequency range input on the EXTAL pin can be used. When the CK output is in use, the frequency range is from 32 to 40 MHz. When an input signal on the EXTAL pin is in use, the XTAL pin should be left open.

Table 4.3 Relationship between Clock Operating Mode and Frequency Range

Clock Operating Mode	FRQCR Setting	PLL Frequency Multiplier		Ratio of Internal Clock Frequencies	Selectable Frequency Range (MHz)				
		PLL Circuit 1	PLL Circuit 2	(I:B:P)※1	Input Clock※2	Output Clock (CK Pin)	Internal Clock (Iϕ)	Bus Clock (Bϕ)	Peripheral Clock (Pϕ)
6	H'1000	On (× 1)	On (×4)	4:4:4	8 to 10	32 to 40	32 to 40	32 to 40	32 to 40
	H'1001	On (× 1)	On (×4)	4:4:2	8 to 10	32 to 40	32 to 40	32 to 40	16 to 20
	H'1003	On (× 1)	On (×4)	4:4:1	8 to 10	32 to 40	32 to 40	32 to 40	8 to 10
	H'1005	On (× 1)	On (×4)	4:4:1/2	8 to 10	32 to 40	32 to 40	32 to 40	4 to 5
	H'1101	On (× 2)	On (×4)	8:4:4	8 to 10	32 to 40	64 to 80	32 to 40	32 to 40
	H'1103	On (× 2)	On (×4)	8:4:2	8 to 10	32 to 40	64 to 80	32 to 40	16 to 20
	H'1105	On (× 2)	On (×4)	8:4:1	8 to 10	32 to 40	64 to 80	32 to 40	8 to 10
	H'1111	On (× 2)	On (×4)	4:4:4	8 to 10	32 to 40	32 to 40	32 to 40	32 to 40
	H'1113	On (× 2)	On (×4)	4:4:2	8 to 10	32 to 40	32 to 40	32 to 40	16 to 20
	H'1115	On (× 2)	On (×4)	4:4:1	8 to 10	32 to 40	32 to 40	32 to 40	8 to 10
	H'1303	On (× 4)	On (×4)	16:4:4	8 to 10	32 to 40	128 to 160	32 to 40	32 to 40
	H'1305	On (× 4)	On (×4)	16:4:2	8 to 10	32 to 40	128 to 160	32 to 40	16 to 20
	H'1313	On (× 4)	On (×4)	8:4:4	8 to 10	32 to 40	64 to 80	32 to 40	32 to 40
	H'1315	On (× 4)	On (×4)	8:4:2	8 to 10	32 to 40	64 to 80	32 to 40	16 to 20
	H'1333	On (× 4)	On (×4)	4:4:4	8 to 10	32 to 40	32 to 40	32 to 40	32 to 40
	H'1335	On (× 4)	On (×4)	4:4:2	8 to 10	32 to 40	32 to 40	32 to 40	16 to 20

Notes: 1. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.

2. The frequency of the clock input from the EXTAL pin or the frequency of the crystal resonator.

Caution: 1. The frequency of the internal clock (Iϕ) is the frequency of the signal input to the CK pin after multiplication by the frequency-multiplier of PLL circuit 1 and division by the divider's divisor. Set the frequency of the internal clock to 160 MHz or less but not less than the frequency of the signal on the CK pin.

2. The frequency of the peripheral clock (Pϕ) is the frequency of the signal input to the CK pin after multiplication by the frequency-multiplier of PLL circuit 1 and division by the divider's divisor. Set the frequency of the peripheral clock to 40 MHz or less. In addition, do not set a higher frequency for the internal clock than the frequency on the CK pin.

3. The frequency multiplier of PLL circuit 1 can be selected as ×1, ×2, or ×4. The divisor of the divider can be selected as ×1, ×1/2, ×1/4, or ×1/8. The settings are made in the frequency-control register (FRQCR).

4. The signal output by PLL circuit 1 is the signal on the CK pin multiplied by the frequency multiplier of PLL circuit 1. Ensure that the frequency of the signal from PLL circuit 1 is no more than 160 MHz.

4.4 Register Descriptions

The clock pulse generator has the following registers.

Table 4.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Frequency control register	FRQCR	R/W	H'1003	H'FFFE0010	16
MTU2S clock frequency control register	MCLKCR	R/W	H'43	H'FFFE0410	8
AD clock frequency control register	ACLKCR	R/W	H'43	H'FFFE0414	8

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). Only word access can be used on FRQCR.

FRQCR is initialized to H'1003 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.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	CKOEN	-	-	STC[1:0]		-	IFC[2:0]			RNGS	PFC[2:0]		
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R/W	R	R	R/W	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
15 to 13	—	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
12	CKOEN	1	R/W	Clock Output Enable Specifies whether a clock is output on the CK pin, or the CK pin is placed in the level-fixed state during software standby mode or when exiting software standby mode. If this bit is cleared to 0, the CK pin is fixed to the low level during software standby mode or when exiting software standby mode. Therefore, the malfunction of an external circuit because of an unstable CK clock upon exit from software standby mode can be prevented. 0: The CK pin is fixed to the low level during software standby mode or when exiting software standby mode. 1: Clock is output from the CK pin (placed in the high-impedance state during software standby mode).
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	STC[1:0]	00	R/W	Frequency multiplication ratio of PLL circuit 1 00: × 1 time 01: × 2 times 10: Setting prohibited 11: × 4 times
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	IFC[2:0]	000	R/W	Internal Clock (Iϕ) Frequency Division Ratio These bits specify the frequency division ratio of the internal clock with respect to the output frequency of PLL circuit 1. If a prohibited value is specified, correct operation cannot be guaranteed. 000: × 1 time 001: × 1/2 time 011: × 1/4 time Other than above: Setting prohibited
3	RNGS	0	R/W	Set this bit according to the output frequency of PLL circuit 1. 0: High-frequency mode 1: Low-frequency mode Always specify high-frequency mode for this LSI. Do not set this bit to 1.
2 to 0	PFC[2:0]	011	R/W	Peripheral Clock (Pϕ) Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock with respect to the output frequency of PLL circuit 1. If a prohibited value is specified, correct operation cannot be guaranteed. 000: × 1 time 001: × 1/2 time 011: × 1/4 time 101: × 1/8 time Other than above: Setting prohibited

4.4.2 MTU2S Clock Frequency Control Register (MCLKCR)

MCLKCR is an 8-bit readable/writable register. Only byte access can be used on MCLKCR.

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
	MSSCS[1:0]		-	-	-	-	MSDIVS[1:0]	
Initial value:	0	1	0	0	0	0	1	1
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MSSCS[1:0]	01	R/W	Source Clock Select These bits select the source clock. 00: Clock stop 01: PLL1 output clock 10: Reserved (setting prohibited) 11: Reserved (setting prohibited)
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	MSDIVS[1:0]	11	R/W	Division Ratio Select These bits specify the frequency division ratio of the source clock. Set these bits so that the output clock is 80 MHz or less, and also an integer multiple of the peripheral clock frequency (Pϕ). 00: × 1 time 01: × 1/2 time 10: Setting prohibited 11: × 1/4 time

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 only initialized to H'43 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
	ASSCS[1:0]		-	-	-	-	ASDIVS[1:0]	
Initial value:	0	1	0	0	0	0	1	1
R/W:	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7, 6	ASSCS[1:0]	01	R/W	Source Clock Select These bits select the source clock. 00: Clock stoppage 01: PLL1 output clock 10: Reserved (setting prohibited) 11: Reserved (setting prohibited)
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	ASDIVS[1:0]	11	R/W	Division Ratio Select These bits specify the frequency division ratio of the source clock. Set these bits so that the output clock is 40 MHz or less, and also an integer multiple of the peripheral clock frequency (Pϕ). 00: × 1 time 01: × 1/2 time 10: Setting prohibited 11: × 1/4 time

4.5 Changing the Frequency

The frequency of the internal clock (Iϕ) and peripheral clock (Pϕ) can be changed either by changing the multiplication rate of PLL circuit 1 or by changing the division rates of divider. All of these are controlled by software through the frequency control register (FRQCR). The methods are described below.

4.5.1 Changing the Multiplication Rate

A PLL settling time is required when the multiplication rate of PLL circuit 1 is changed. When the multiplication rate is changed, the LSI temporarily stops automatically and the internal watchdog timer (WDT) starts counting the settling time. When the count of the WDT overflows, the LSI restarts operating with the set clock frequency. The following shows this setting procedure.

1. In the initial state, the multiplication rate of PLL circuit 1 is 1 time.
2. Set a value that will become the specified oscillation settling time in the WDT and stop the WDT. The following must be set:

   WTCSR.TME = 0: WDT stops

   WTCSR.CKS[2:0]: Division ratio of WDT count clock

   WTCNT counter: Initial counter value

   For setting of the counter, determine the overflow period with the frequency after the peripheral clock (Pϕ) setting change.
3. Set the desired value in the STC[1:0] bits. The division ratio can also be set in the IFC[2:0] and PFC[2:0] bits.
4. This LSI pauses temporarily and the WDT starts incrementing. The internal and peripheral clocks both stop and the WDT is supplied with the clock. The clock will continue to be output at the CK pin. This state is the same as software standby mode. Whether or not registers are initialized depends on the module. For details, see table 23.4 in section 23, Power-Down Modes.
5. Supply of the clock that has been set begins at WDT count overflow, and this LSI begins operating again. The WDT stops counting after it overflows.

4.5.2 Changing the Division Ratio

Counting by the WDT does not proceed if the frequency divisor is changed but the multiplier is not.

1. In the initial state, IFC[2:0] = B'000 and PFC[2:0] = B'011.
2. Set the desired value in the IFC[2:0] and PFC[2:0] bits. The values that can be set are limited by the clock operating mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, this LSI will malfunction.
3. After the register bits (IFC[2:0] and PFC[2:0]) have been set, the clock is supplied of the new division ratio.

Note: When executing the SLEEP instruction after the frequency has been changed, be sure to read the frequency control register (FRQCR) three times before executing the SLEEP instruction.

4.6 Notes on Board Design

4.6.1 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.

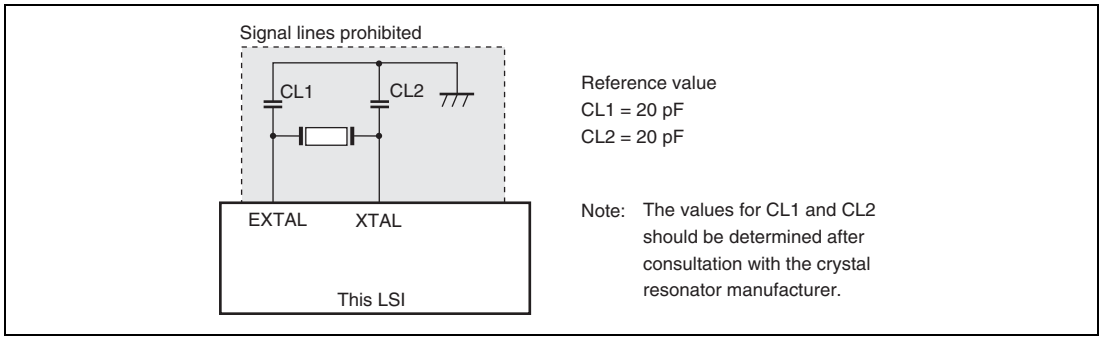


Figure 4.2 Note on Using a Crystal Resonator

4.6.2 Note on Bypass Capacitor

A multilayer ceramic capacitor should be inserted for each pair of Vss and Vcc as a bypass capacitor as many as possible. The bypass capacitor must be inserted as close to the power supply pins of the LSI as possible. Note that the capacitance and frequency characteristics of the bypass capacitor must be appropriate for the operating frequency of the LSI.

4.6.3 Note on Using a PLL Oscillation Circuit

In the PLLVcc and PLLVss connection pattern for the PLL, signal lines from the board power supply pins must be as short as possible and pattern width must be as wide as possible to reduce inductive interference.

Since the analog power supply pins of the PLL are sensitive to the noise, the system may malfunction due to inductive interference at the other power supply pins. To prevent such malfunction, the analog power supply pin Vcc and digital power supply pin VccQ should not supply the same resources on the board if at all possible.

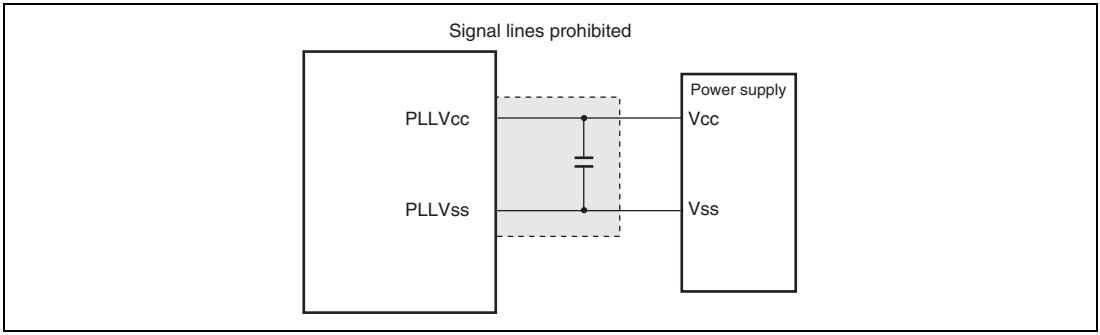


Figure 4.3 Note on Using a PLL Oscillation Circuit

Section 5 Exception Handling

5.1 Overview

5.1.1 Types of Exception Handling and Priority

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

Table 5.1 Types of Exception Handling and Priority Order

Type	Exception Handling		Priority
Reset	Power-on reset		High
	Manual reset		
Address error	CPU address error		
	DMAC address error		
Instruction	Integer division exception (division by zero)		
	Integer division exception (overflow)		
Register bank error	Bank underflow		
	Bank overflow		
Interrupt	NMI		
	User break		
	H-UDI		
	IRQ		
	On-chip peripheral modules	A/D converter (ADC)	
		Direct memory access controller (DMAC)	
		Compare match timer (CMT)	
		Bus state controller (BSC)	
		Watchdog timer (WDT)	
		Multi-function timer pulse unit 2 (MTU2)	
		Port output enable 2 (POE2): OEI1 and OEI2 interrupts	Low

Type	Exception Handling		Priority
Interrupt	On-chip peripheral modules	Multi-function timer pulse unit 2S (MTU2S)	High
		Port output enable 2 (POE2): OEI3 interrupt	
		I^2C bus interface 3 (IIC3)	
		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, MOV.B@disp12, MOV.W@disp12, MOV.L@disp12, MOVI20, MOVI20S, MOVU.B, MOVU.W.

5.1.2 Exception Handling Operations

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

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

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

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 number 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
(Reserved by system)		13	H'00000034 to H'00000037
H-UDI		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043

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

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

Table 5.4 Calculating Exception Handling Vector Table Addresses

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

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

5.2 Resets

5.2.1 Types of Reset

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

Table 5.5 Exception Source Detection and Exception Handling Start Timing

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

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

5.2.2 Power-On Reset

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

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

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

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

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

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

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

(3) Power-On Reset Initiated by WDT

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

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

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

5.2.3 Manual Reset

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

When the $\overline{\text{MRES}}$ pin is driven low, this LSI enters the manual reset state. To reset this LSI without fail, the $\overline{\text{MRES}}$ pin should be kept at the low level for at least 20-tcyc. 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. However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than the fixed internal manual reset interval cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception handling is not executed.

5.3 Address Errors

5.3.1 Address Error Sources

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

Table 5.6 Bus Cycles and Address Errors

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

Note: * See section 8, Bus State Controller (BSC), for details of the on-chip peripheral module space and on-chip RAM space.

5.3.2 Address Error Exception Handling

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

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

Note: * This is the case in which an address error was caused by data read or write. When an address error is caused by an instruction fetch, and if the bus cycle in which the address error occurred does not end by step 3 above, the CPU restarts the address error exception handling until the bus cycle ends.

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, and on-chip peripheral modules.

Table 5.7 Interrupt Sources

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller (UBC)	1
H-UDI	User debugging interface (H-UDI)	1
IRQ	IRQ0 to IRQ7 pins (external input)	8
On-chip peripheral module	A/D converter (ADC)	1
	Direct memory access controller (DMAC)	16
	Compare match timer (CMT)	2
	Bus state controller (BSC)	1
	Watchdog timer (WDT)	1
	Multi-function timer pulse unit 2 (MTU2)	26
	Multi-function timer pulse unit 2S (MTU2S)	13
	Port output enable 2 (POE2)	3
	I^2C bus interface 3 (IIC3)	5
	Serial communication interface with FIFO (SCIF)	16

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 15 (IPR01, IPR02, and IPR05 to IPR15) 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 15 (IPR01, IPR02, IPR05 to IPR15), for details of IPR01, IPR02, and IPR05 to IPR15.

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 01, 02, and 05 to 15 (IPR01, IPR02, and IPR05 to IPR15).
On-chip peripheral module		

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 number 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, general illegal instructions, slot illegal instructions, and integer division exceptions, 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, 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

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.

5.7 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

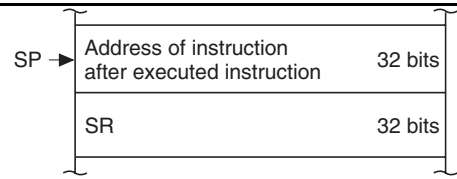
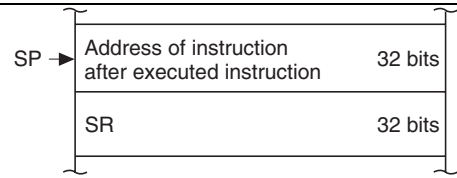
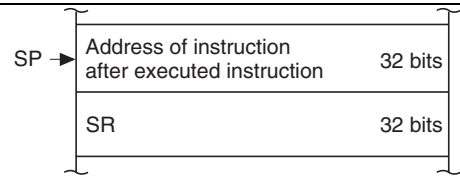
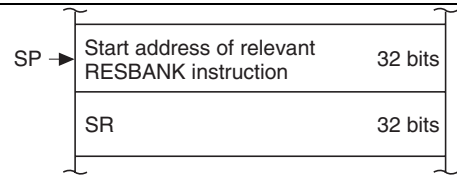
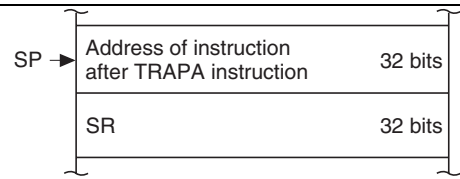
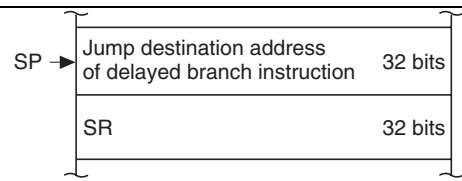
	Exception Source		
Point of Occurrence	**Address Error**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted

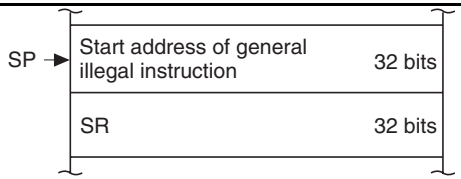
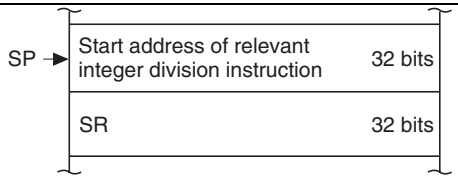
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF

5.8 Stack Status after Exception Handling Ends

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

Table 5.11 Stack Status After Exception Handling Ends

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

Exception Type	Stack Status
General illegal instruction	
Integer division instruction	

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

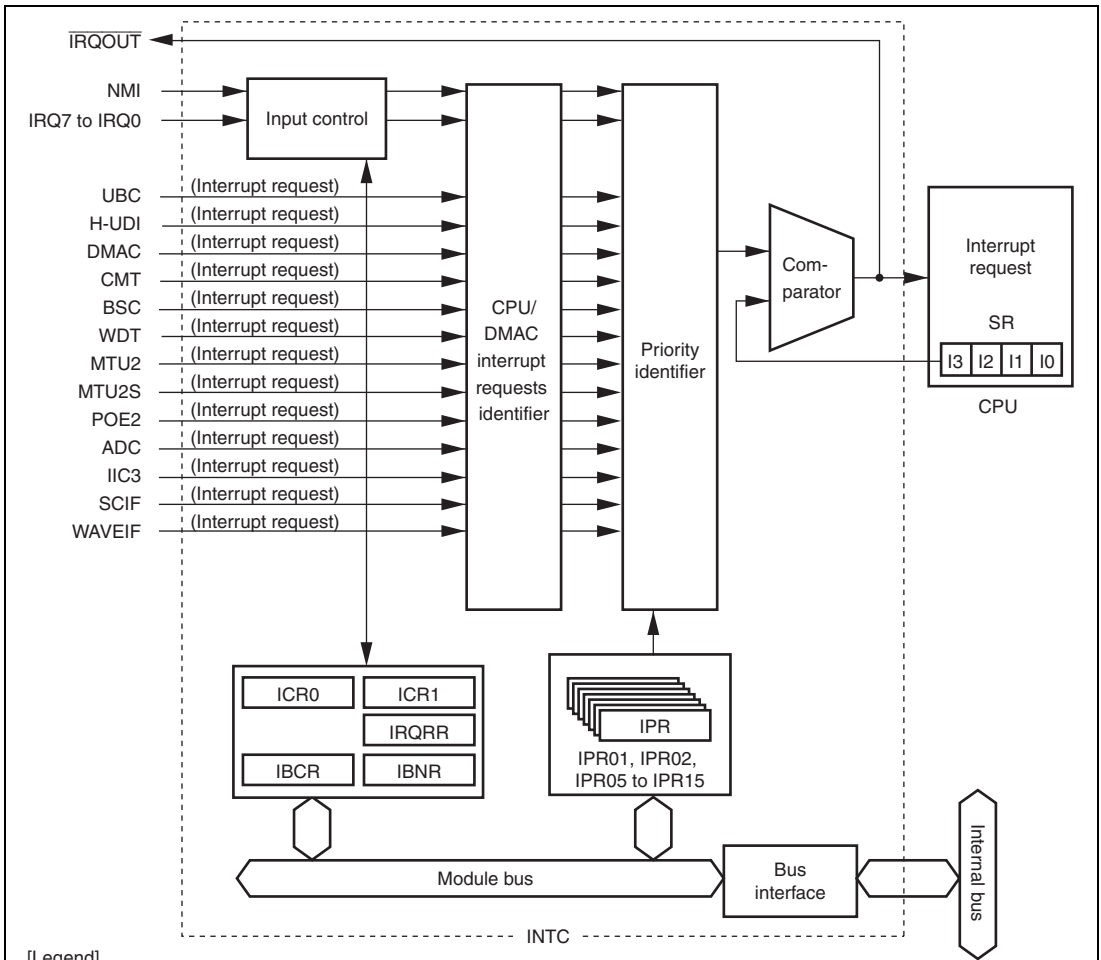
  By setting the thirteen interrupt priority registers, the priorities of IRQ interrupts and on-chip peripheral module interrupts can be selected from 16 levels for request sources.
- NMI noise canceler function

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

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

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

Figure 6.1 shows a block diagram of the INTC.



[Legend]

UBC:	User break controller
H-UDI:	User debugging interface
DMAC:	Direct memory access controller
CMT:	Compare match timer
BSC:	Bus state controller
WDT:	Watchdog timer
MTU2:	Multi-function timer pulse unit 2
MTU2S:	Multi-function timer pulse unit 2S
POE2:	Port output enable 2
ADC:	A/D converter
IIC3:	I^2C bus interface 3
SCIF:	Serial communication interface with FIFO
WAVEIF:	WAVE interface

ICR0:	Interrupt control register 0
ICR1:	Interrupt control register 1
IRQRR:	IRQ interrupt request register
IBCR:	Bank control register
IBNR:	Bank number register
IPR01, 02, 05 to 15:	Interrupt priority registers 01, 02, 05 to 15

Figure 6.1 Block Diagram of INTC

6.2 Input/Output Pins

Table 6.1 shows the pin configuration of the INTC.

Table 6.1 Pin Configuration

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

6.3 Register Descriptions

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

Table 6.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt control register 0	ICR0	R/W	*1	H'FFFE0800	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFE0802	16, 32
IRQ interrupt request register	IRQRR	R/(W)*2	H'0000	H'FFFE0806	16, 32
Bank control register	IBCR	R/W	H'0000	H'FFFE080C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFE080E	16, 32
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, 32
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFE0820	16, 32
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, 32
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, 32
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, 32
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, 32
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, 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 15 (IPR01, IPR02, IPR05 to IPR15)

IPR01, IPR02, and IPR05 to IPR15 are 16-bit readable/writable registers in which priority levels from 0 to 15 are set for IRQ interrupts, PINT 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 IPR15.

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 IPR15

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 01	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register 02	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register 05	Reserved	Reserved	ADI	Reserved
Interrupt priority register 06	DMAC0	DMAC1	DMAC2	DMAC3
Interrupt priority register 07	DMAC4	DMAC5	DMAC6	DMAC7
Interrupt priority register 08	CMT0	CMT1	BSC	WDT
Interrupt priority register 09	MTU0 (TGI0A to TGI0D)	MTU0 (TCI0V, TGI0E, TGI0F)	MTU1 (TGI1A, TGI1B)	MTU1 (TCI1V, TCI1U)
Interrupt priority register 10	MTU2 (TGI2A, TGI2B)	MTU2 (TCI2V, TCI2U)	MTU3 (TGI3A to TGI3D)	MTU3 (TCI3V)
Interrupt priority register 11	MTU4 (TGI4A to TGI4D)	MTU4 (TCI4V)	MTU5 (TGI5U, TGI5V, TGI5W)	POE2 (OEI1, OEI2)
Interrupt priority register 12	MTU3S (TGI3A to TGI3D)	MTU3S (TCI3V)	MTU4S (TGI4A to TGI4D)	MTU4S (TCI4V)

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 13	MTU5S (TGI5U, TGI5V, TGI5W)	POE2 (OEI3)	IIC3	Reserved
Interrupt priority register 14	SCIF0	SCIF1	SCIF2	SCIF3
Interrupt priority register 15	WAVEIF	Reserved	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 IPR15 are initialized to H'0000 by a power-on reset.

6.3.2 Interrupt Control Register 0 (ICR0)

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

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

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

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

6.3.3 Interrupt Control Register 1 (ICR1)

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

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

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ71S	0	R/W	IRQ Sense Select These bits select whether interrupt signals corresponding to pins IRQ7 to IRQ0 are detected by a low level, falling edge, rising edge, or both edges. 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
14	IRQ70S	0	R/W	
13	IRQ61S	0	R/W	
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 = 7 to 0

6.3.4 IRQ Interrupt Request Register (IRQRR)

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

IRQRR is initialized by a power-on reset.

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

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
7	IRQ7F	0	R/(W)*	IRQ Interrupt Request These bits indicate the status of the IRQ7 to IRQ0 interrupt requests. Level detection: 0: IRQn interrupt request has not occurred [Clearing condition] • IRQn input is high 1: IRQn interrupt has occurred [Setting condition] • IRQn input is low Edge detection: 0: IRQn interrupt request is not detected [Clearing conditions] • Cleared by reading IRQnF while IRQnF = 1, then writing 0 to IRQnF • Cleared by executing IRQn interrupt exception handling 1: IRQn interrupt request is detected [Setting condition] • Edge corresponding to IRQn1S or IRQn0S of ICR1 has occurred at IRQn pin
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

[Legend]

n = 7 to 0

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

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
14	E14	0	R/W	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.
13	E13	0	R/W	
12	E12	0	R/W	0: Use of register banks is disabled
11	E11	0	R/W	1: Use of register banks is enabled
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

6.3.6 Bank Number Register (IBNR)

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

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

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

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

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

6.4 Interrupt Sources

There are five types of interrupt sources: NMI, user break, H-UDI, IRQ, 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 24, User Debugging Interface (H-UDI).

6.4.4 IRQ Interrupts

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

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

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

The IRQ interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the accepted IRQ interrupt.

6.4.5 On-Chip Peripheral Module Interrupts

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

- A/D converter (ADC)
- Direct memory access controller (DMAC)
- Compare match timer (CMT)
- Bus state controller (BSC)
- Watchdog timer (WDT)
- Multi-function timer pulse unit 2 (MTU2)
- Multi-function timer pulse unit 2S (MTU2S)
- Port output enable 2 (POE2)
- I^2C bus interface 3 (IIC3)
- Serial communication interface with FIFO (SCIF)
- WAVE interface (WAVEIF)

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 15 (IPR05 to IPR15). 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 15 (IPR01, IPR02, and IPR05 to IPR15). However, if two or more interrupts specified by the same IPR among IPR05 to IPR15 occur, the priorities are defined as shown in the IPR setting unit internal priority of table 6.4, and the priorities cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, they are processed by the default priorities indicated in table 6.4.

Table 6.4 Interrupt Exception Handling Vectors and Priorities

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

Interrupt Source		Interrupt Vector Number	Interrupt Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
DMAC	DMAC0	DEI0	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR06 (15 to 12)	1	High
		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	
	DMAC6	DEI6	132	H'00000210 to H'00000213	0 to 15 (0)	IPR07 (7 to 4)	1	
		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	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
CMT	CMI0		140	H'00000230 to H'00000233	0 to 15 (0)	IPR08 (15 to 12)	—	High
	CMI1		144	H'00000240 to H'00000243	0 to 15 (0)	IPR08 (11 to 8)	—	
BSC	CMI		148	H'00000250 to H'00000253	0 to 15 (0)	IPR08 (7 to 4)	—	
WDT	ITI		152	H'00000260 to H'00000263	0 to 15 (0)	IPR08 (3 to 0)	—	
MTU2	MTU0	TGI0A	156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (15 to 12)	1	
		TGI0B	157	H'00000274 to H'00000277			2	
		TGI0C	158	H'00000278 to H'0000027B			3	
		TGI0D	159	H'0000027C to H'0000027F			4	
		TCI0V	160	H'00000280 to H'00000283	0 to 15 (0)	IPR09 (11 to 8)	1	
		TGI0E	161	H'00000284 to H'00000287			2	
		TGI0F	162	H'00000288 to H'0000028B			3	
	MTU1	TGI1A	164	H'00000290 to H'00000293	0 to 15 (0)	IPR09 (7 to 4)	1	
		TGI1B	165	H'00000294 to H'00000297			2	
		TCI1V	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPR09 (3 to 0)	1	
		TCI1U	169	H'000002A4 to H'000002A7			2	Low

Interrupt Source		Interrupt Vector Number	Interrupt Vector	Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU2	TGI2A	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPR10 (15 to 12)	1	High
		TGI2B	173	H'000002B4 to H'000002B7			2	
		TCI2V	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPR10 (11 to 8)	1	
		TCI2U	177	H'000002C4 to H'000002C7			2	
	MTU3	TGI3A	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR10 (7 to 4)	1	
		TGI3B	181	H'000002D4 to H'000002D7			2	
		TGI3C	182	H'000002D8 to H'000002DB			3	
		TGI3D	183	H'000002DC to H'000002DF			4	
		TCI3V	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR10 (3 to 0)	—	
	MTU4	TGI4A	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR11 (15 to 12)	1	
		TGI4B	189	H'000002F4 to H'000002F7			2	
		TGI4C	190	H'000002F8 to H'000002FB			3	
		TGI4D	191	H'000002FC to H'000002FF			4	
		TCI4V	192	H'00000300 to H'00000303	0 to 15 (0)	IPR11 (11 to 8)	—	
	MTU5	TGI5U	196	H'00000310 to H'00000313	0 to 15 (0)	IPR11 (7 to 4)	1	
		TGI5V	197	H'00000314 to H'00000317			2	
		TGI5W	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	MTU3S	TGI3A	204	H'00000330 to H'00000333	0 to 15 (0)	IPR12 (15 to 12)	1	
		TGI3B	205	H'00000334 to H'00000337			2	
		TGI3C	206	H'00000338 to H'0000033B			3	
		TGI3D	207	H'0000033C to H'0000033F			4	
		TCI3V	208	H'00000340 to H'00000343	0 to 15 (0)	IPR12 (11 to 8)	—	
	MTU4S	TGI4A	212	H'00000350 to H'00000353	0 to 15 (0)	IPR12 (7 to 4)	1	
		TGI4B	213	H'00000354 to H'00000357			2	
		TGI4C	214	H'00000358 to H'0000035B			3	
		TGI4D	215	H'0000035C to H'0000035F			4	
		TCI4V	216	H'00000360 to H'00000363	0 to 15 (0)	IPR12 (3 to 0)	—	
	MTU5S	TGI5U	220	H'00000370 to H'00000373	0 to 15 (0)	IPR13 (15 to 12)	1	
		TGI5V	221	H'00000374 to H'00000377			2	
		TGI5W	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
IIC3	STPI		228	H'00000390 to H'00000393	0 to 15 (0)	IPR13 (7 to 4)	1	High
	NAKI		229	H'00000394 to H'00000397			2	
	RXI		230	H'00000398 to H'0000039B			3	
	TXI		231	H'0000039C to H'0000039F			4	
	TEI		232	H'000003A0 to H'000003A3			5	
SCIF	SCIF0	BRI0	240	H'000003C0 to H'000003C3	0 to 15 (0)	IPR14 (15 to 12)	1	
		ERI0	241	H'000003C4 to H'000003C7			2	
		RXI0	242	H'000003C8 to H'000003CB			3	
		TXI0	243	H'000003CC to H'000003CF			4	
	SCIF1	BRI1	244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR14 (11 to 8)	1	
		ERI1	245	H'000003D4 to H'000003D7			2	
		RXI1	246	H'000003D8 to H'000003DB			3	
		TXI1	247	H'000003DC to H'000003DF			4	
	SCIF2	BRI2	248	H'000003E0 to H'000003E3	0 to 15 (0)	IPR14 (7 to 4)	1	
		ERI2	249	H'000003E4 to H'000003E7			2	
		RXI2	250	H'000003E8 to H'000003EB			3	
		TXI2	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	
WAVEIF	ERR		256	H'00000400 to H'00000403	0 to 15 (0)	IPR15 (15 to 12)	1	
	WRXI		257	H'00000404 to H'00000407			2	
	WTXI		258	H'00000408 to H'0000040B			3	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 15 (IPR01, IPR02, and IPR05 to IPR15). 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.

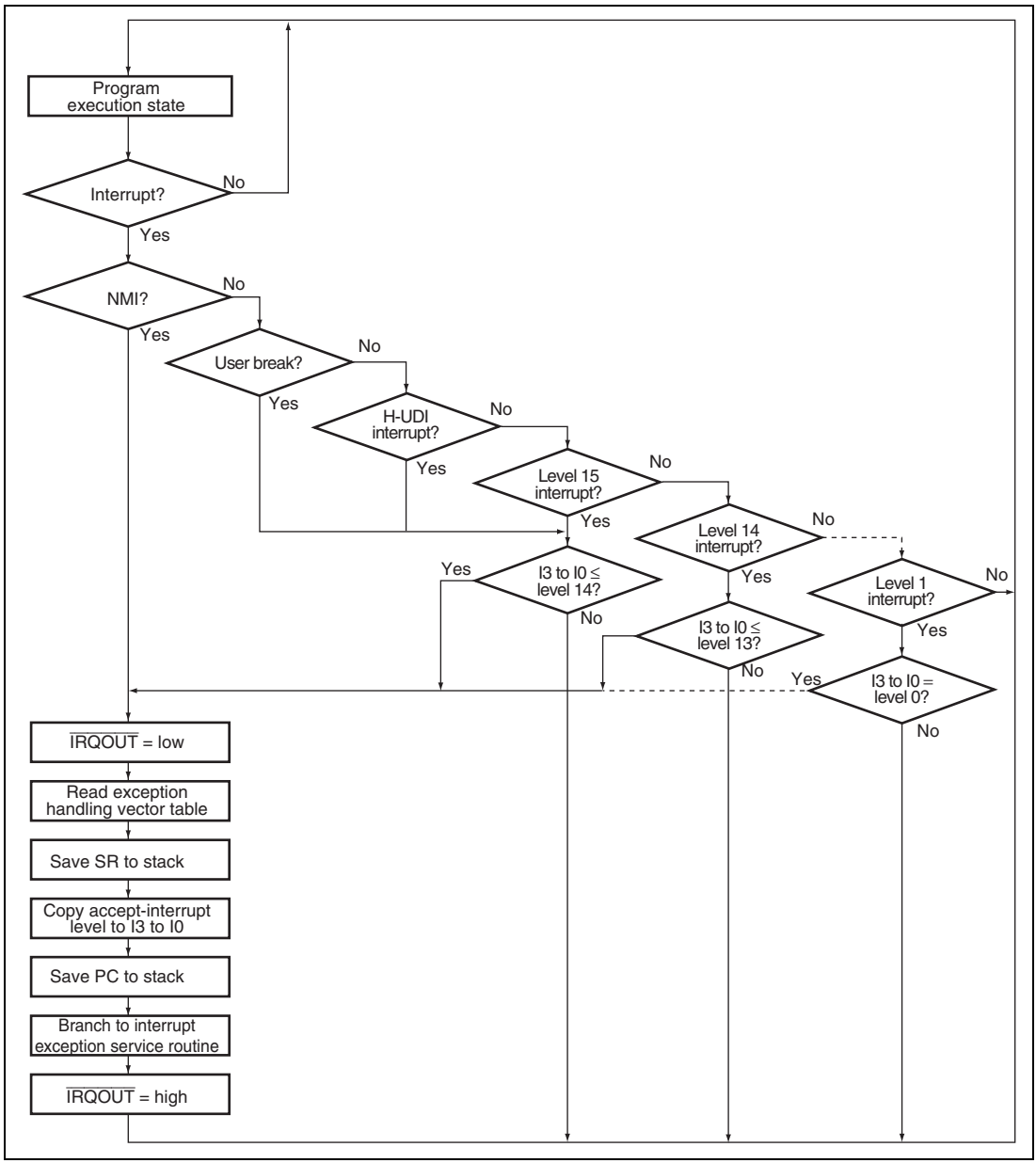
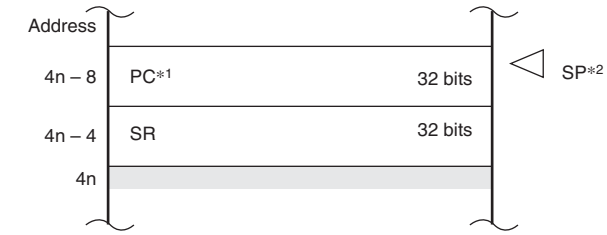


Figure 6.2 Interrupt Operation Flow

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**	**User Break**	**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 + 1 Pcyc	
Time from input of interrupt request signal to CPU until sequence currently being executed is completed, interrupt exception handling starts, and first instruction in interrupt exception service routine is fetched	No register banking	Min.	3 Icyc + m1 + m2					Min. is when the interrupt wait time is zero. Max. is when a higher-priority interrupt request has occurred during interrupt exception handling.
		Max.	4 Icyc + 2(m1 + m2) + m3					
	Register banking without register bank overflow	Min.	—		3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—		12 Icyc + m1 + m2			
	Register banking with register bank overflow	Min.	—		3 Icyc + m1 + m2			Min. is when the interrupt wait time is zero. Max. is when an interrupt request has occurred during execution of the RESBANK instruction.
		Max.	—		3 Icyc + m1 + m2 + 19(m4)			

Item			Number of States					Remarks
			NMI	User Break	H-UDI	IRQ	Peripheral Module	
Interrupt response time	No register banking	Min.	5 Icyc + 2 Bcyc + 1 Pcyc + m1 + m2	6 Icyc + m1 + m2	5 Icyc + 1 Pcyc + m1 + m2	5 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	5 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	160-MHz operation*1*2: 0.050 to 0.106 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	160-MHz operation*1*2: 0.075 to 0.131 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	160-MHz operation*1*2: 0.050 to 0.106 µs
		Max.	—	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	160-MHz operation*1*2: 0.106 to 0.163 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	160-MHz operation*1*2: 0.050 to 0.106 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)	160-MHz operation*1*2: 0.169 to 0.225 µs

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

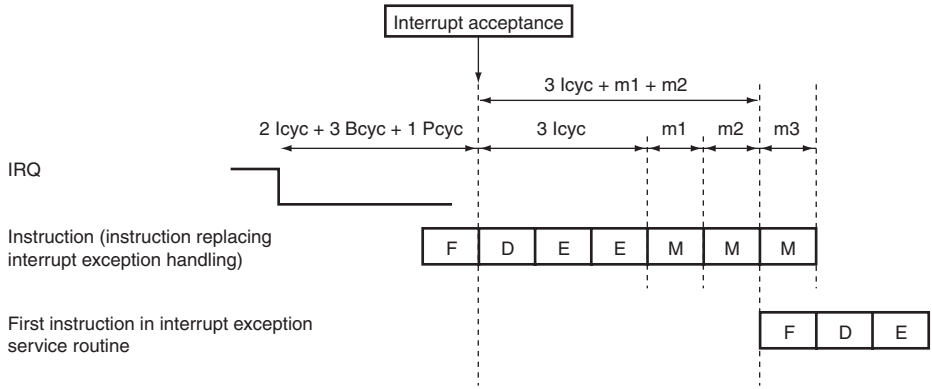
m1: Vector address read (longword read)

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) = (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)

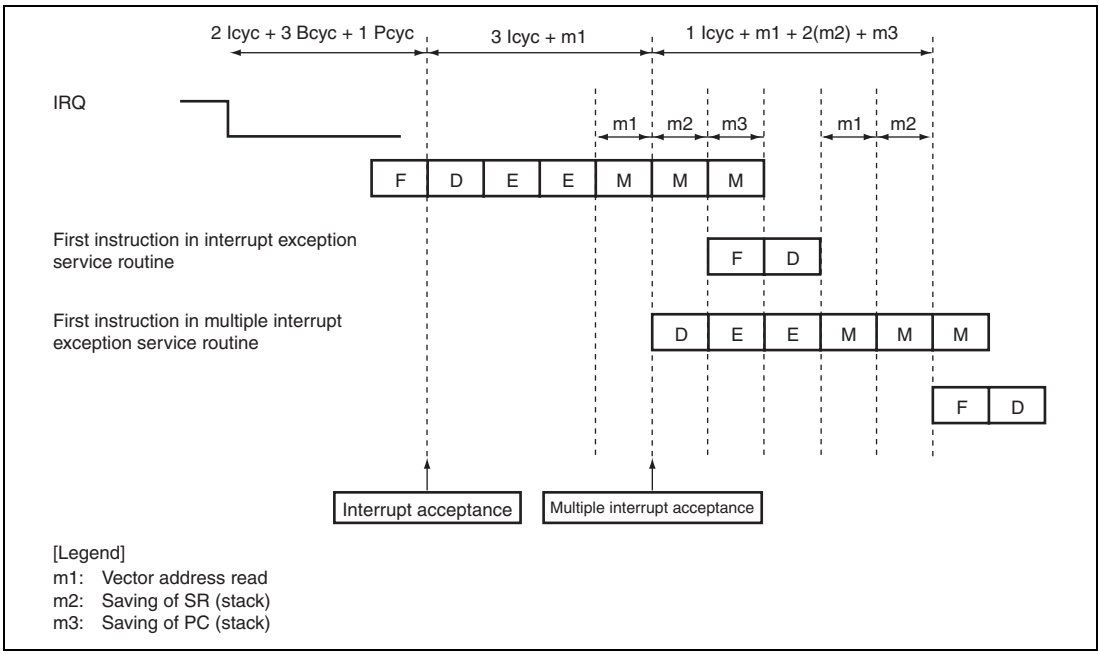


Figure 6.5 Example of Pipeline Operation for Multiple Interrupts (No Register Banking)

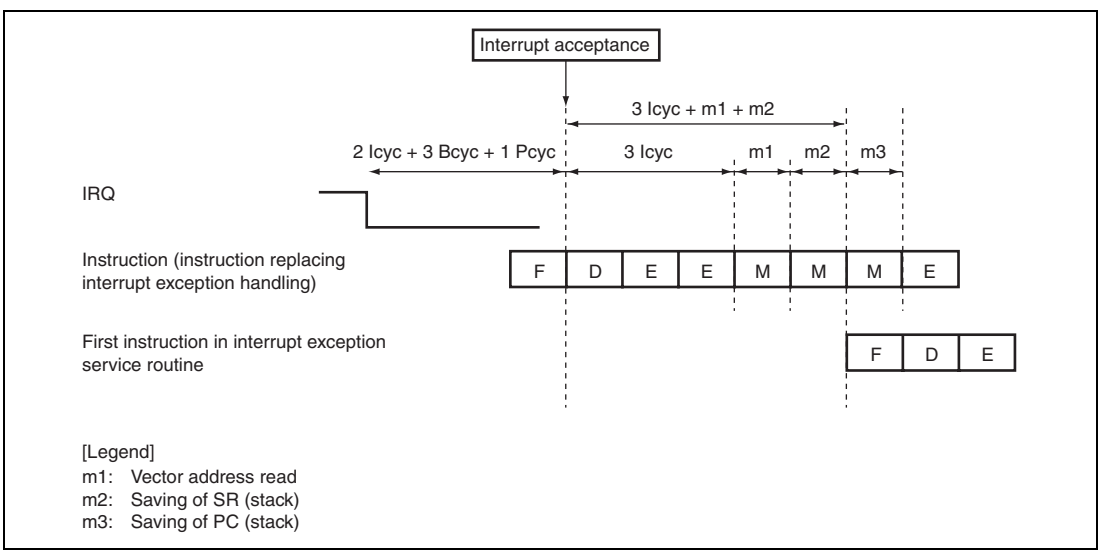


Figure 6.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

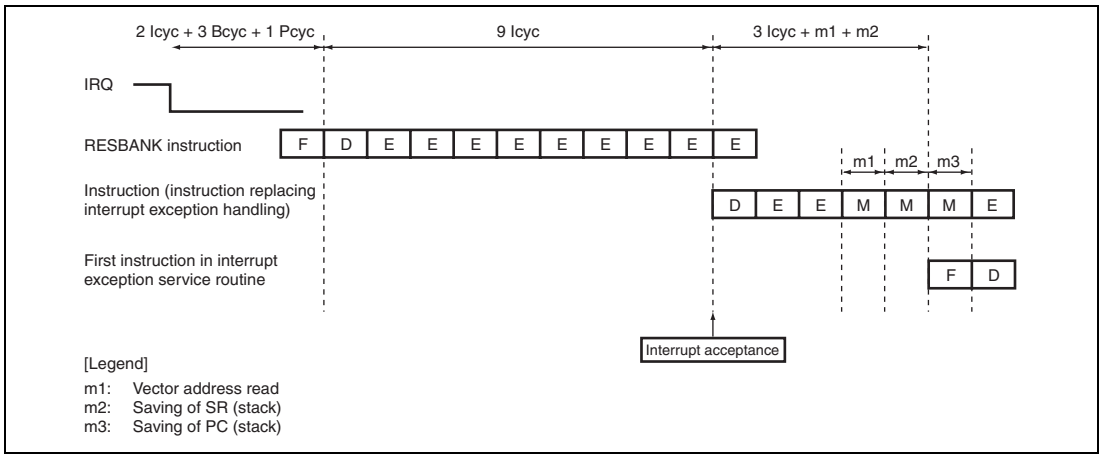


Figure 6.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

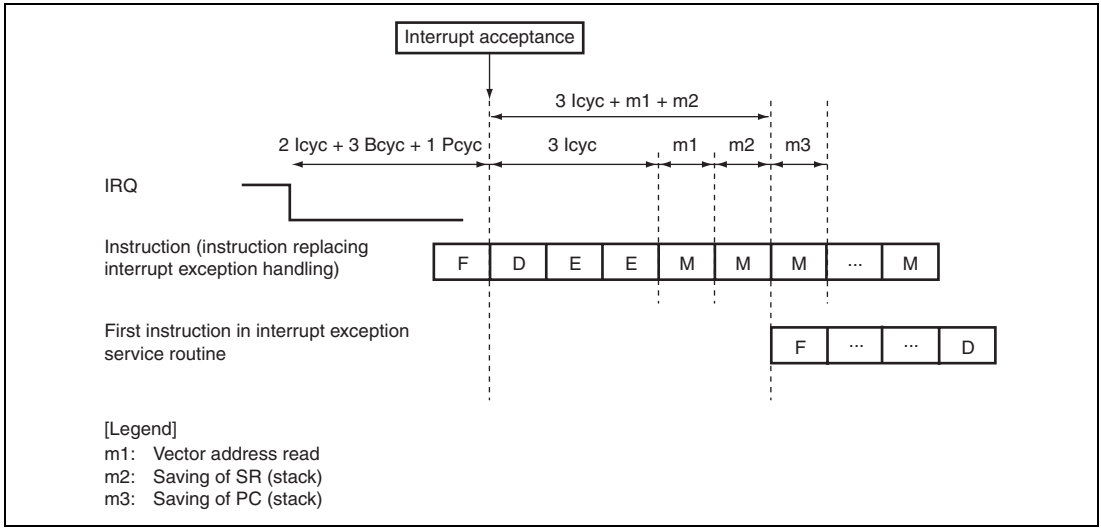


Figure 6.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

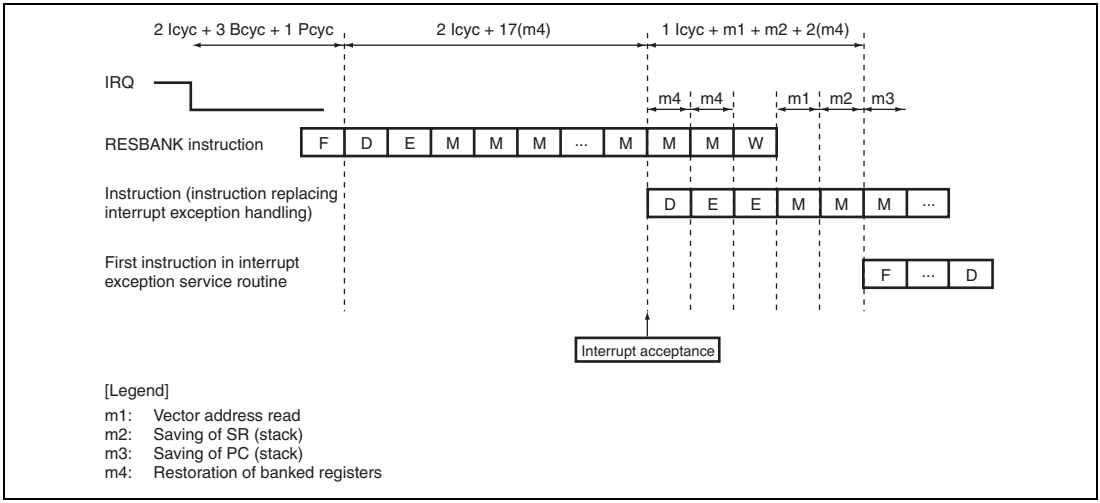


Figure 6.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

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.

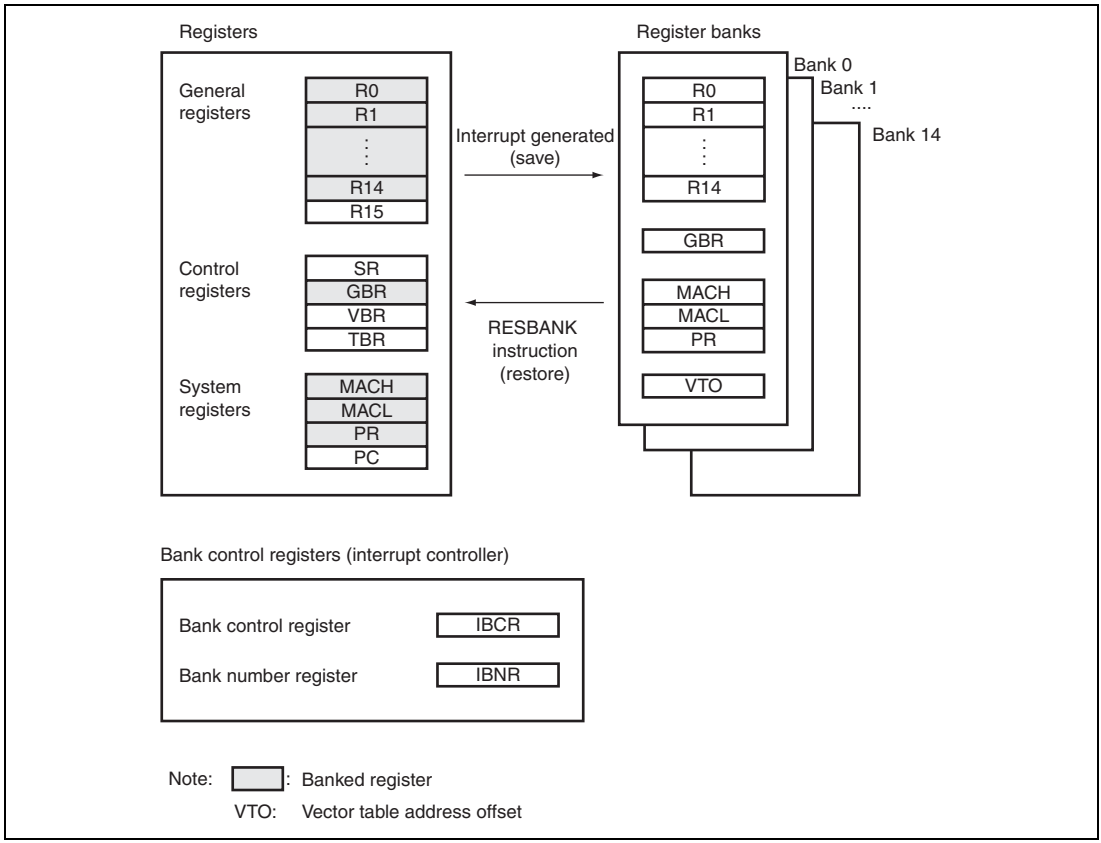


Figure 6.10 Overview of Register Bank Configuration

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.

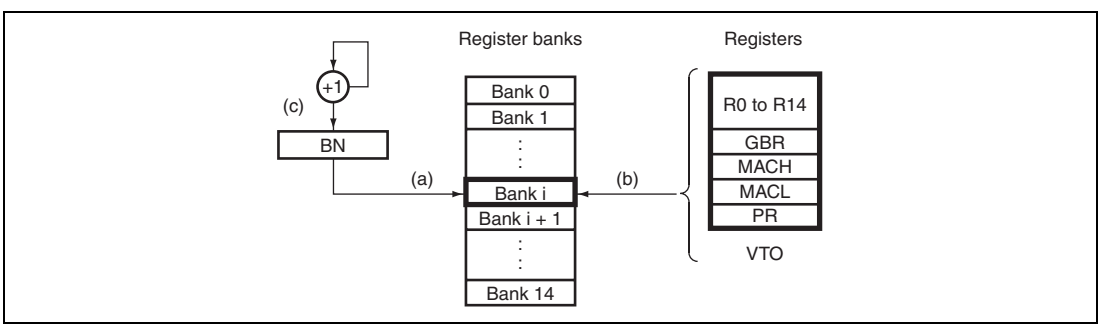


Figure 6.11 Bank Save Operations

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.

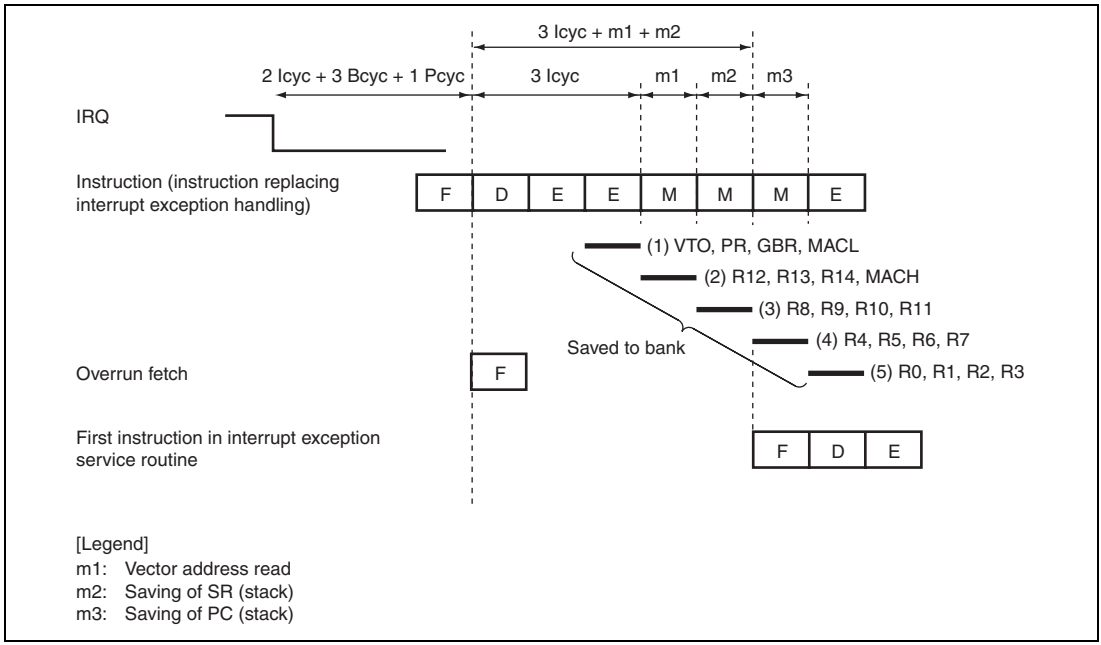


Figure 6.12 Bank Save Timing

(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 Data Transfer with Interrupt Request Signals

Interrupt request signals can be used to activate the DMAC and transfer data.

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)

Figure 6.13 shows a block diagram of interrupt control.

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 9, Direct Memory Access Controller (DMAC).

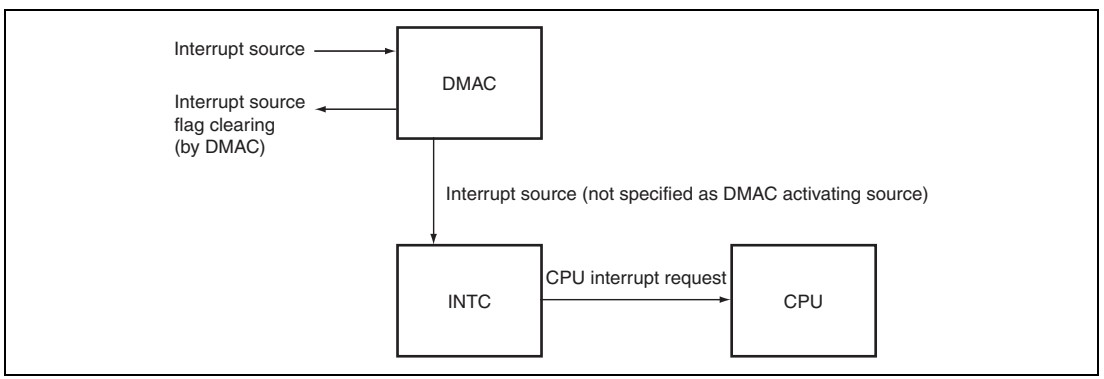


Figure 6.13 Interrupt Control Block Diagram

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

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. When interrupts occur, interrupt requests are sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt exception service routine.

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

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 settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears the interrupt sources when starting transfer.

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, and then execute an RTE instruction.

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 or DMAC) 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 or DMAC 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.

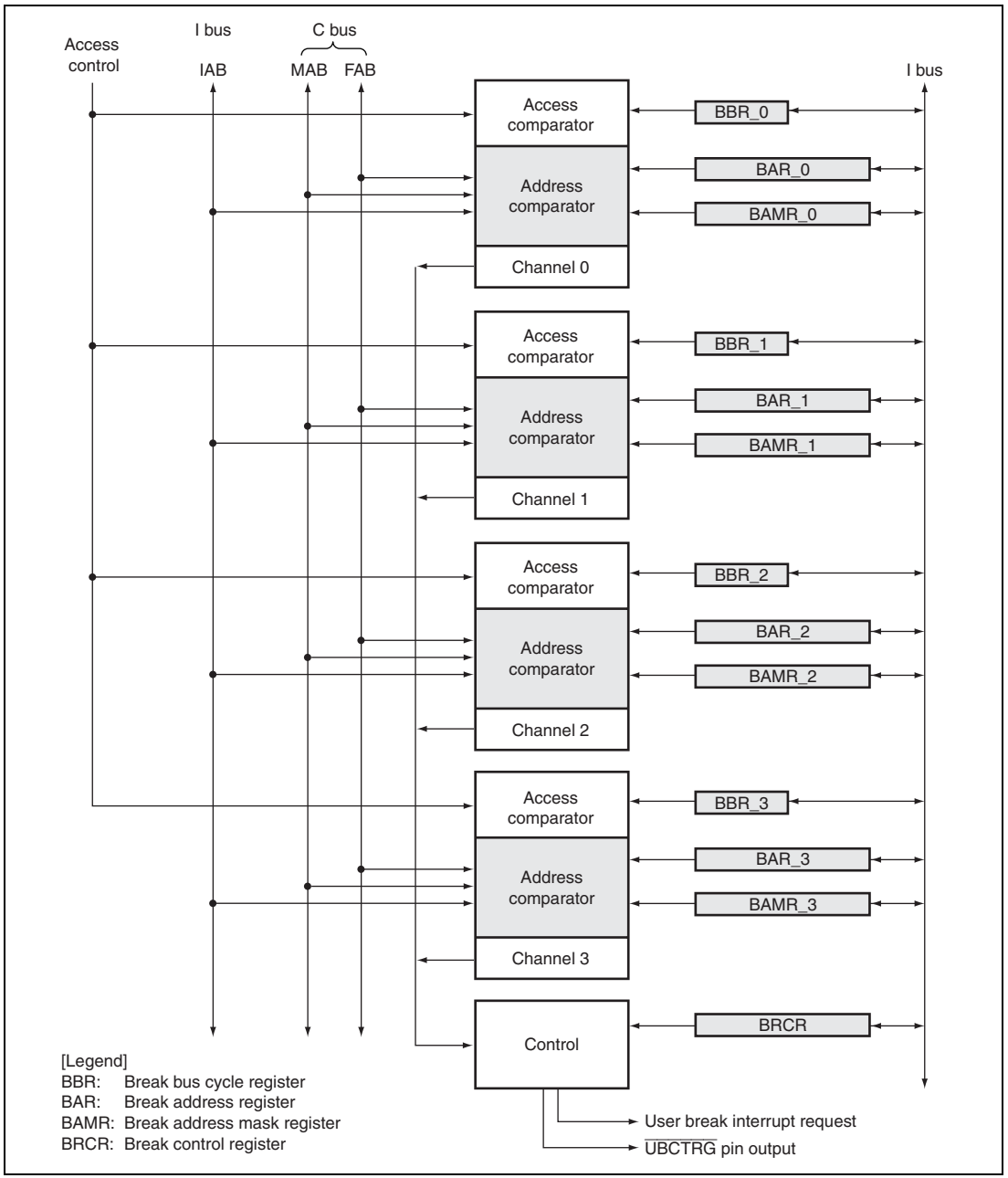


Figure 7.1 Block Diagram of UBC

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[1:0]		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	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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	CP0[1:0]	00	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). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break conditions

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

[Legend]

x: Don't care

7.3.4 Break Address Register_1 (BAR_1)

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

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

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

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

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

7.3.5 Break Address Mask Register_1 (BAMR_1)

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

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

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

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

Note: n = 31 to 0

7.3.6 Break Bus Cycle Register_1 (BBR_1)

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID1	-	-	-	CP1[1:0]		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	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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	CP1[1:0]	00	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). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break conditions

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

[Legend]

x: Don't care

7.3.7 Break Address Register_2 (BAR_2)

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

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

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

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

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

7.3.8 Break Address Mask Register_2 (BAMR_2)

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

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

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

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

Note: n = 31 to 0

7.3.9 Break Bus Cycle Register_2 (BBR_2)

BBR_2 is a 16-bit readable/writable register, which specifies (1) disabling or enabling of user break interrups, (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[1:0]		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	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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	CP2[1:0]	00	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). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break conditions

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

[Legend]

x: Don't care

7.3.10 Break Address Register_3 (BAR_3)

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

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

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

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

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

7.3.11 Break Address Mask Register_3 (BAMR_3)

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

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

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

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

Note: n = 31 to 0

7.3.12 Break Bus Cycle Register_3 (BBR_3)

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID3	-	-	-	CP3[1:0]		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	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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	CP3[1:0]	00	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). x1: CPU cycle is included in break conditions 1x: DMAC cycle is included in break conditions

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

[Legend]

x: Don't care

7.3.13 Break Control Register (BRCR)

BRCR sets the following conditions:

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

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

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

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

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

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

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

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

7.4 Operation

7.4.1 Flow of the User Break Operation

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

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

— I bus cycles (including read fill cycles) resulting from instruction fetches on the C bus by the CPU are defined as instruction fetch cycles on the I bus, while other bus cycles are defined as data access cycles.

— The DMAC only issues data access cycles for I bus cycles.

— If a break condition is specified for the I bus, even when the condition matches in an I bus cycle resulting from an instruction executed by the CPU, at which instruction the user break is to be accepted cannot be clearly defined.

7.4.2 Break on Instruction Fetch Cycle

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

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

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

7.4.3 Break on Data Access Cycle

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

Table 7.3 Data Access Cycle Addresses and Operand Size Comparison Conditions

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

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

Longword access at H'00001000

Word access at H'00001002

Byte access at H'00001003

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

7.4.4 Value of Saved Program Counter

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

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

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

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

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

7.4.5 Usage Examples

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

(Example 1-1)

- Register specifications

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

  <Channel 0>

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

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

  <Channel 1>

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

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

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

(Example 1-2)

- Register specifications

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

  <Channel 0>

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

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

  <Channel 1>

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

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

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

(Example 1-3)

- Register specifications

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

  <Channel 0>

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

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

  <Channel 1>

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

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

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

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

(Example 2-1)

- Register specifications

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

  <Channel 0>

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

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

  <Channel 1>

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

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

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

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

(Example 3-1)

- Register specifications

  BBR_0 = H'0094, BAR_0 = H'00314156, BAMR_0 = H'00000000,
  BBR_1 = H'12A9, BAR_1 = H'00055555, BAMR_1 = H'00000000, BRCR = H'00000000

  <Channel 0>

  Address: H'00314156, Address mask: H'00000000

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

  <Channel 1>

  Address: H'00055555, Address mask: H'00000000

  Bus cycle: I bus/data access/write/byte

  On channel 0, the setting of I bus/instruction fetch is ignored.

  On channel 1, a user break occurs when the DMAC writes byte data in address H'00055555 on the I bus (write by the CPU does not generate a user break).

7.5 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 interrupt request and another exception source 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 source with higher priority occurs, the user break interrupt request is not received.
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 user break interrupt request is not received 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 user break before instruction execution for the instruction following the DIVU or DIVS instruction. If a user break before instruction execution is set for the instruction following the DIVU or DIVS instruction and an exception or interrupt occurs during execution of the DIVU or DIVS instruction, a user break occurs before instruction execution even though execution of the DIVU or DIVS instruction is halted.
10. Do not set a user break both before instruction execution and after instruction execution for instruction of the same address. If, for example, a user break before instruction execution on channel 0 and a user break after instruction on channel 1 are set at the instruction of the same address, the condition match flag for the channel 1 is set even though a user break on channel 0 occurs before instruction execution.

Section 8 Bus State Controller (BSC)

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

8.1 Features

The BSC has the following features.

1. External address space
   - A maximum of 64 Mbytes for each of areas CS0 to CS7.
   - Can specify the normal space interface, SRAM interface with byte selection, burst ROM (clock synchronous or asynchronous), MPX-I/O, and SDRAM for each address space.
   - Can select the data bus width (8 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. Burst ROM interface (clock asynchronous)
   - High-speed access to the ROM that has the page mode function.
4. MPX-I/O interface
   - Can directly connect to a peripheral LSI that needs an address/data multiplexing.
5. SDRAM interface
   - Can set the SDRAM in up to two areas.
   - Multiplex output for row address/column address.
   - Efficient access by single read/single write.
   - High-speed access in bank-active mode.
   - Supports an auto-refresh and self-refresh.
   - Supports low-frequency and power-down modes.
   - Issues MRS and EMRS commands.

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

Figure 8.1 shows a block diagram of the BSC.

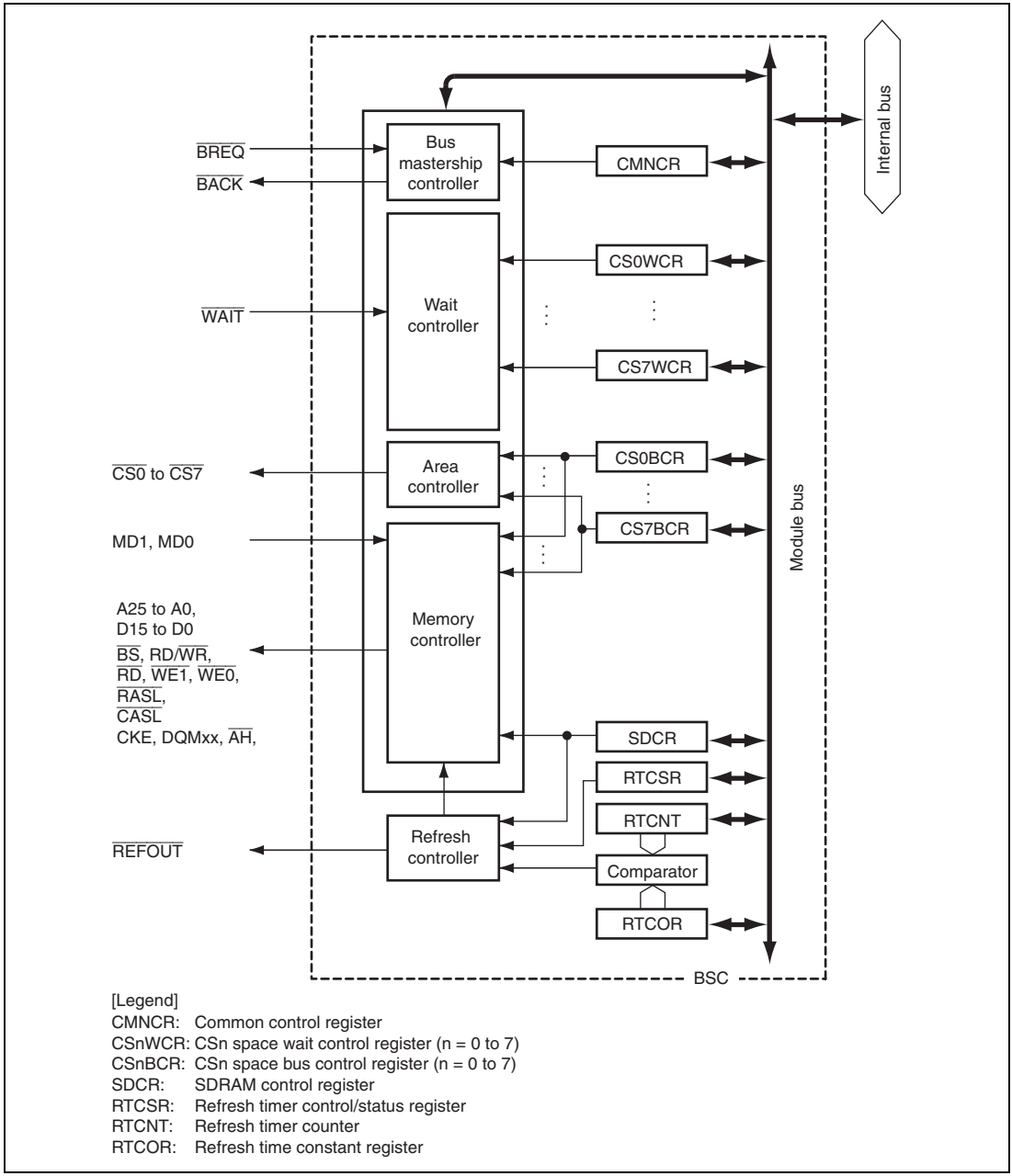


Figure 8.1 Block Diagram of BSC

8.2 Input/Output Pins

Table 8.1 shows the pin configuration of the BSC.

Table 8.1 Pin Configuration

Name	I/O	Function
A25 to A0	Output	Address bus
D15 to D0	I/O	Data bus
$\overline{BS}$	Output	Bus cycle start
$\overline{CS0}$ to $\overline{CS7}$	Output	Chip select
RD/$\overline{WR}$	Output	Read/write Connects to $\overline{WE}$ pins when SDRAM or SRAM with byte selection is connected.
$\overline{RD}$	Output	Read pulse signal (read data output enable signal) Functions as a strobe signal for indicating memory read cycles when PCMCIA is used.
$\overline{AH}$	Output	A signal used to hold an address when MPX-I/O is in use
$\overline{WE1}$/$\overline{DQMLU}$	Output	Indicates that D15 to D8 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D15 to D8 when SDRAM is connected.
$\overline{WE0}$/$\overline{DQMLL}$	Output	Indicates that D7 to D0 are being written to. Connected to the byte select signal when a SRAM with byte selection is connected. Functions as the select signals for D7 to D0 when SDRAM is connected.
$\overline{RASL}$	Output	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{CASL}$	Output	Connects to $\overline{CAS}$ pin when SDRAM is connected.
CKE	Output	Connects to CKE pin when SDRAM is connected.
$\overline{WAIT}$	Input	External wait input
$\overline{BREQ}$	Input	Bus request input
$\overline{BACK}$	Output	Bus enable output
$\overline{REFOUT}$	Output	Refresh request output in bus-released state
MD1, MD0	Input	Select bus width (8 or 16 bits) of area 0 and modes including enabling/disabling of the on-chip ROM.

8.3 Area Overview

8.3.1 Address Map

In the architecture, this LSI has a 32-bit address space, which is divided into external address space and on-chip spaces (on-chip ROM, on-chip RAM, on-chip peripheral modules, and reserved areas) according to the upper bits of the address.

The kind of memory to be connected and the data bus width are specified in each partial space. The address map for the external address space is listed below.

Table 8.2 Address Map in On-Chip ROM-Enabled Mode

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'0007 FFFF	On-chip ROM	On-chip ROM	512 Kbytes
H'0008 0000 to H'01FF FFFF	Other	Reserved area	—
H'0200 0000 to H'03FF FFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)	32 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM, reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 22, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 26, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

Table 8.3 Address Map in On-Chip ROM-Disabled Mode

Address	Space	Memory to be Connected	Size
H'0000 0000 to H'03FF FFFF	CS0	Normal space, SRAM with byte selection, burst ROM (asynchronous or synchronous)	64 Mbytes
H'0400 0000 to H'07FF FFFF	CS1	Normal space, SRAM with byte selection	64 Mbytes
H'0800 0000 to H'0BFF FFFF	CS2	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'0C00 0000 to H'0FFF FFFF	CS3	Normal space, SRAM with byte selection, SDRAM	64 Mbytes
H'1000 0000 to H'13FF FFFF	CS4	Normal space, SRAM with byte selection, burst ROM (asynchronous)	64 Mbytes
H'1400 0000 to H'17FF FFFF	CS5	Normal space, SRAM with byte selection, MPX-I/O	64 Mbytes
H'1800 0000 to H'1BFF FFFF	CS6	Normal space, SRAM with byte selection	64 Mbytes
H'1C00 0000 to H'1FFF FFFF	CS7	Normal space, SRAM with byte selection	64 Mbytes
H'2000 0000 to H'FFF7 FFFF	Other	Reserved area	—
H'FFF8 0000 to H'FFFB FFFF	Other	On-chip RAM, reserved area*	—
H'FFFC 0000 to H'FFFF FFFF	Other	On-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 22, On-Chip RAM. For the on-chip I/O register space, access the addresses shown in section 26, List of Registers. Do not access addresses which are not described in these sections. Otherwise, the correct operation cannot be guaranteed.

8.3.2 Setting Operating Modes

This LSI can set the following modes of operation at the time of power-on reset using the external pins.

- Single-Chip Mode

  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.

- On-Chip ROM-Enabled Mode/On-Chip ROM-Disabled Mode

  In on-chip ROM-enabled mode, since the first half of area 0 is allocated to the on-chip ROM, the LSI can be activated by the on-chip ROM program upon a power-on reset. The second half of area 0 is the external memory space.

In on-chip ROM-disabled mode, the LSI is activated by the program stored in the external memory allocated to area 0. The second half of area 0 is the external memory space. In this case, a ROM is assumed for the external memory of area 0. Therefore, minimum functions are provided for the pins including address bus, data bus, CS0, and RD. Although BS, RDWR, WEn, and other pins are shown in the examples of access waveforms in this section, these are examples when pin settings are performed by the pin function controller. For details, see section 19, Pin Function Controller (PFC). Do not perform any operation except for area 0 read access until the pin settings by the program is completed.

- Initial Settings of Data Bus Widths for Areas 0 to 7

  The initial settings of data bus widths of areas 0 to 7 can be selected at a time as 8 or 16 bits.

  In on-chip ROM-disabled mode, the data bus width of area 0 cannot be changed from its initial setting after a power-on reset, but the data bus widths of areas 1 to 7 can be changed by register settings in the program. In on-chip ROM-enabled mode, all the data bus widths of areas 0 to 7 can be changed by register settings in the program. Note that data bus widths will be restricted depending on memory types.

- Initial Settings of Endianness

  The initial settings of byte-data alignment of areas 0 to 7 can be selected as big endian or little endian. In on-chip ROM-disabled mode, the endianness of area 0 cannot be changed from its initial setting after a power-on reset, but the endianness of areas 1 to 7 can be changed by register settings in the program. In on-chip ROM-enabled mode, all the endianness of areas 0 to 7 can be changed by register settings in the program. Little endian cannot be selected in area 0. Since both 32-bit and 16-bit accesses are included in instruction fetches, no instructions can be assigned in little endian area. Accordingly, instructions should be executed in big endian area.

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

8.4 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 8.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Common control register	CMNCR	R/W	H'00001010	H'FFFC0000	32
CSn space bus control register	CSnBCR	R/W	H'36DB0400 (in activation with 16-bit bus width)	H'FFFC 0004 to H'FFFC 0020	32
CSn space wait control register	CSnWCR	R/W	H'00000500	H'FFFC0028 to H'FFFC 0044	32
SDRAM control register	SDCR	R/W	H'00000000	H'FFFC004C	32
Refresh timer control/status register	RTCSR	R/W	H'00000000	H'FFFC0050	32
Refresh timer counter	RTCNT	R/W	H'00000000	H'FFFC0054	32
Refresh time constant register	RTCOR	R/W	H'00000000	H'FFFC0058	32

8.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the common items for each area. This register is initialized to H'00001010 by a power-on reset and retains the value by a manual reset and in software standby mode.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	BLOCK	DPRTY[1:0]		DMAIW[2:0]			DMA IWA	-	-	HIZ CKIO	HIZ MEM	HIZ CNT
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
11	BLOCK	0	R/W	Bus Lock Specifies whether or not the $\overline{\text{BREQ}}$ signal is received. 0: Receives $\overline{\text{BREQ}}$. 1: Does not receive $\overline{\text{BREQ}}$.
10, 9	DPRTY[1:0]	00	R/W	DMA Burst Transfer Priority Specify the priority for a refresh request/bus mastership request during DMA burst transfer. 00: Accepts a refresh request and bus mastership request during DMA burst transfer. 01: Accepts a refresh request but does not accept a bus mastership request during DMA burst transfer. 10: Accepts neither a refresh request nor a bus mastership request during DMA burst transfer. 11: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
8 to 6	DMAIW[2:0]	000	R/W	Wait states between access cycles when DMA single address transfer is performed. Specify the number of idle cycles to be inserted after an access to an external device with DACK when DMA single address transfer is performed. The method of inserting idle cycles depends on the contents of DMAIWA. 000: No idle cycle inserted 001: 1 idle cycle inserted 010: 2 idle cycles inserted 011: 4 idle cycles inserted 100: 6 idle cycles inserted 101: 8 idle cycles inserted 110: 10 idle cycles inserted 111: 12 idle cycles inserted
5	DMAIWA	0	R/W	Method of inserting wait states between access cycles when DMA single address transfer is performed. Specifies the method of inserting the idle cycles specified by the DMAIW[2:0] bit. Clearing this bit will make this LSI insert the idle cycles when another device, which includes this LSI, drives the data bus after an external device with DACK drove it. However, when the external device with DACK drives the data bus continuously, idle cycles are not inserted. Setting this bit will make this LSI insert the idle cycles after an access to an external device with DACK, even when the continuous access cycles to an external device with DACK are performed. 0: Idle cycles inserted when another device drives the data bus after an external device with DACK drove it. 1: Idle cycles always inserted after an access to an external device with DACK
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	HIZCKIO	0	R/W	High-Z CK Control Specifies the state in CK standby mode and when bus mastership is released. 0: CK is in high impedance state in standby mode and bus-released state. 1: CK is driven in standby mode and bus-released state.
1	HIZMEM	0	R/W	High-Z Memory Control Specifies the pin state in standby mode for A25 to A0, $\overline{BS}$, $\overline{CS}$n, RD/$\overline{WR}$, $\overline{WE}$n/DQMxx, $\overline{AH}$, and $\overline{RD}$. At bus-released state, these pins are in high-impedance state regardless of the setting value of the HIZMEM bit. 0: High impedance in standby mode. 1: Driven in standby mode
0	HIZCNT	0	R/W	High-Z Control Specifies the state in standby mode and bus-released state for CKE, $\overline{RASL}$, and $\overline{CASL}$. 0: CKE, $\overline{RASL}$, and $\overline{CASL}$ are in high-impedance state in standby mode and bus-released state. 1: CKE, $\overline{RASL}$, and $\overline{CASL}$ are driven in standby mode and bus-released state.

8.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7)

CSnBCR is a 32-bit readable/writable register that specifies the type of memory connected to a space, data bus width of an area, endian, and the number of waits between access cycles. This register is initialized to H'36DB0x00 by a power-on reset and retains the value by a manual reset and in software standby mode.

Do not access external memory other than area 0 until CSnBCR initial setting is completed.

Idle cycles may be inserted even when they are not specified. For details, see section 8.5.10, Wait between Access Cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Initial value:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TYPE[2:0]			ENDIAN	BSZ[1:0]		-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1*	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

Note: * CSnBCR samples the external pins (MD1 and MD0) that specify the bus width at power-on reset.

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.

Bit	Bit Name	Initial Value	R/W	Description
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
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

Bit	Bit Name	Initial Value	R/W	Description
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
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 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
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.
14 to 12	TYPE[2:0]	000	R/W	Specify the type of memory connected to a space. 000: Normal space 001: Burst ROM (clock asynchronous) 010: MPX-I/O 011: SRAM with byte selection 100: SDRAM 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Burst ROM (clock synchronous) For details of memory type in each area, see tables 8.2 and 8.3.
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]	11*	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 Notes: 1. If area 5 is specified as MPX-I/O, the bus width can be specified as 8 bits or 16 bits by the address according to the SZSEL bit in CS5WCR by specifying the BSZ[1:0] bits to 11. The fixed bus width can be specified as 8 bits or 16 bits. 2. The initial data bus width for areas 0 to 7 is specified by external pins. In on-chip ROM-disabled mode, writing to the BSZ1 and BSZ0 bits in CS0BCR is ignored, but the bus width settings in CS1BCR to CS7BCR can be modified. In on-chip ROM-enabled mode, the bus width settings in CS0BCR to CS7BCR can be modified. 3. If area 2 or area 3 is specified as SDRAM space, the bus width can be specified as 16 bits only. 4. If area 0 or 4 is specified as clock-synchronous burst ROM space, the bus width can be specified as 16 bits only.
8 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * CSnBCR samples the external pins (MD1 and MD0) that specify the bus width at power-on reset.

8.4.3 CSn Space Wait Control Register (CSnWCR) (n = 0 to 7)

CSnWCR specifies various wait cycles for memory access. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

CSnWCR is initialized to H'00000500 by a power-on reset and retains the value by a manual reset and in software standby mode.

(1) Normal Space, SRAM with Byte Selection, MPX-I/O

- CS0WCR

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	— *	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS*	0	R/W	Byte Access Selection when SRAM with Byte Selection is Used Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.

Bit	Bit Name	Initial Value	R/W	Description
19 to 13	— *	All 0	R/W	Reserved Set these bits to 0 when the interface for normal space or SRAM with byte selection is used.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS0}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address and $\overline{CS0}$ assertion to $\overline{RD}$ and $\overline{WEn}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{WEn}$ Negation to Address, $\overline{CS0}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ negation to address and $\overline{CS0}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

Note * To connect the burst ROM to the CS0 space and switch to the burst ROM interface after activation in ROM-disabled mode, set the TYPE[2:0] bits in CS0BCR after setting the burst number by the bits 20 and 21 and the burst wait cycle number by the bits16 and 17. Do not write 1 to the reserved bits other than above bits.

- CS1WCR, CS7WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read/write timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CSn}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address and $\overline{CSn}$ assertion to $\overline{RD}$ and $\overline{WEn}$ 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 RD, $\overline{\text{WEn}}$ Negation to Address, $\overline{\text{CSn}}$ Negation Specify the number of delay cycles from RD and $\overline{\text{WEn}}$ negation to address and $\overline{\text{CSn}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS2WCR, CS3WCR

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	WR[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of cycles that are necessary for read/write access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	WR[3:0]	1010	R/W	Number of Read Access Wait Cycles Specify the number of cycles that are necessary for read access. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from RD, $\overline{\text{WEn}}$ Negation to Address, $\overline{\text{CS4}}$ Negation Specify the number of delay cycles from RD and $\overline{\text{WEn}}$ negation to address and $\overline{\text{CS4}}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

- CS5WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SZSEL	MPXW/ BAS	-	WW[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21	SZSEL	0	R/W	MPX-I/O Interface Bus Width Specification Specifies an address to select the bus width when the BSZ[1:0] of CS5BCR are specified as 11. This bit is valid only when area 5 is specified as MPX-I/O. 0: Selects the bus width by address A14 1: Selects the bus width by address A21 The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized below.

SZSEL	A14	A21	Bus Width
0	0	Not affected	8 bits
0	1	Not affected	16 bits
1	Not affected	0	8 bits
1	Not affected	1	16 bits

Bit	Bit Name	Initial Value	R/W	Description
20	MPXW	0	R/W	MPX-I/O Interface Address Wait This bit setting is valid only when area 5 is specified as MPX-I/O. Specifies the address cycle insertion wait for MPX-I/O interface. 0: Inserts no wait cycle 1: Inserts 1 wait cycle
	BAS	0	R/W	SRAM with Byte Selection Byte Access Select This bit setting is valid only when area 5 is specified as SRAM with byte selection. Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	WW[2:0]	000	R/W	Number of Write Access Wait Cycles Specify the number of cycles that are necessary for write access. 000: The same cycles as WR[3:0] setting (number of read access wait cycles) 001: No cycle 010: 1 cycle 011: 2 cycles 100: 3 cycles 101: 4 cycles 110: 5 cycles 111: 6 cycles
15 to 13	—	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
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS5}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS5}$ assertion to $\overline{RD}$ and $\overline{WE}$ 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
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 RD, $\overline{WEn}$ Negation to Address, $\overline{CS5}$ Negation Specify the number of delay cycles from RD and $\overline{WEn}$ 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
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read/write access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS6}$ Assertion to $\overline{RD}$, $\overline{WEn}$ Assertion Specify the number of delay cycles from address, $\overline{CS6}$ assertion to $\overline{RD}$ and $\overline{WEn}$ 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	WN	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification of this bit is valid even when the number of access wait cycles is 0. 0: The external wait input is valid 1: The external wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Number of Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS6}$ Negation Specify the number of delay cycles from $\overline{RD}$, $\overline{WEn}$ negation to address, and $\overline{CS6}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(2) Burst ROM (Clock Asynchronous)

- CS0WCR

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11.

Bus Width	BST[1:0]	Burst count
8 bits	00	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst

19, 18	—	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
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

- CS4WCR

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BST[1:0]	00	R/W	Burst Count Specification Specify the burst count for 16-byte access. These bits must not be set to B'11.
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bus Width	BST[1:0]	Burst count
8 bits	00	16 burst × one time
	01	4 burst × four times
16 bits	00	8 burst × one time
	01	2 burst × four times
	10	4-4 or 2-4-2 burst

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12, 11	SW[1:0]	00	R/W	Number of Delay Cycles from Address, $\overline{CS4}$ Assertion to $\overline{RD}$, $\overline{WE}$ Assertion Specify the number of delay cycles from address and $\overline{CS4}$ assertion to $\overline{RD}$ and $\overline{WE}$ assertion. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)

Bit	Bit Name	Initial Value	R/W	Description
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	HW[1:0]	00	R/W	Delay Cycles from $\overline{RD}$, $\overline{WEn}$ Negation to Address, $\overline{CS4}$ Negation Specify the number of delay cycles from $\overline{RD}$ and $\overline{WEn}$ negation to address and $\overline{CS4}$ negation. 00: 0.5 cycles 01: 1.5 cycles 10: 2.5 cycles 11: 3.5 cycles

(3) SDRAM*

- CS2WCR

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	A2CL[1:0]		-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A2CL[1:0]	10	R/W	CAS Latency for Area 2 Specify the CAS latency for area 2. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: * If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

- CS3WCR

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	WTRP[1:0]*		-	WTRCD[1:0]*		-	A3CL[1:0]		-	-	TRWL[1:0]*		-	WTRC[1:0]*	
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14, 13	WTRP[1:0]*	00	R/W	Number of Auto-Precharge Completion Wait Cycles Specify the number of minimum precharge completion wait cycles as shown below. • From the start of auto-precharge and issuing of ACTV command for the same bank • From issuing of the PRE/PALL command to issuing of the ACTV command for the same bank • Till entering power-down mode or deep power-down mode • From the issuing of PALL command to issuing REF command in auto-refresh mode • From the issuing of PALL command to issuing SELF command in self-refresh mode The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles

Bit	Bit Name	Initial Value	R/W	Description
12	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
11, 10	WTRCD[1:0]	01	R/W	Number of Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command Specify the minimum number of wait cycles from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8, 7	A3CL[1:0]	10	R/W	CAS Latency for Area 3 Specify the CAS latency for area 3. 00: 1 cycle 01: 2 cycles 10: 3 cycles 11: 4 cycles
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	TRWL[1:0]*	00	R/W	Number of Auto-Precharge Startup Wait Cycles Specify the number of minimum auto-precharge startup wait cycles as shown below. • Cycle number from the issuance of the WRITA command by this LSI until the completion of auto-precharge in the SDRAM. Equivalent to the cycle number from the issuance of the WRITA command until the issuance of the ACTV command. Confirm that how many cycles are required between the WRITE command receive in the SDRAM and the auto-precharge activation, referring to each SDRAM data sheet. And set the cycle number so as not to exceed the cycle number specified by this bit. • Cycle number from the issuance of the WRITA command until the issuance of the PRE command. This is the case when accessing another low address in the same bank in bank active mode. The setting for areas 2 and 3 is common. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
1, 0	WTRC[1:0]*	00	R/W	Number of Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the periods shown below. • From the issuance of the REF command until the issuance of the ACTV/REF/MRS command • From releasing self-refresh until the issuance of the ACTV/REF/MRS command. The setting for areas 2 and 3 is common. 00: 2 cycles 01: 3 cycles 10: 5 cycles 11: 8 cycles

Note: * If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.
If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

(4) Burst ROM (Clock Synchronous)

- CS0WCR

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	BW[1:0]	00	R/W	Number of Burst Wait Cycles Specify the number of wait cycles to be inserted between the second or subsequent access cycles in burst access. 00: No cycle 01: 1 cycle 10: 2 cycles 11: 3 cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 7	W[3:0]	1010	R/W	Number of Access Wait Cycles Specify the number of wait cycles to be inserted in the first access cycle. 0000: No cycle 0001: 1 cycle 0010: 2 cycles 0011: 3 cycles 0100: 4 cycles 0101: 5 cycles 0110: 6 cycles 0111: 8 cycles 1000: 10 cycles 1001: 12 cycles 1010: 14 cycles 1011: 18 cycles 1100: 24 cycles 1101: Reserved (setting prohibited) 1110: Reserved (setting prohibited) 1111: Reserved (setting prohibited)
6	WM	0	R/W	External Wait Mask Specification Specifies whether or not the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

8.4.4 SDRAM Control Register (SDCR)

SDCR specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

SDCR is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	A2ROW[1:0]		-	A2COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV	-	-	-	A3ROW[1:0]		-	A3COL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20, 19	A2ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 2 Specify the number of bits of row address for area 2. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
18	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
17, 16	A2COL[1:0]	00	R/W	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

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	DEEP	0	R/W	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RFSH or RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode
12	SLOW	0	R/W	Low-Frequency Mode Specifies the output timing of command, address, and write data for SDRAM and the latch timing of read data from SDRAM. Setting this bit makes the hold time for command, address, write and read data extended for half cycle (output or read at the falling edge of CK). This mode is suitable for SDRAM with low-frequency clock. 0: Command, address, and write data for SDRAM is output at the rising edge of CK. Read data from SDRAM is latched at the rising edge of CK. 1: Command, address, and write data for SDRAM is output at the falling edge of CK. Read data from SDRAM is latched at the falling edge of CK.
11	RFSH	0	R/W	Refresh Control Specifies whether or not the refresh operation of the SDRAM is performed. 0: No refresh 1: Refresh

Bit	Bit Name	Initial Value	R/W	Description
10	RMODE	0	R/W	Refresh Control Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed
9	PDOWN	0	R	Power-Down Mode Specifies whether the SDRAM will enter power-down mode after the access to the SDRAM. With this bit being set to 1, after the SDRAM is accessed, the CKE signal is driven low and the SDRAM enters power-down mode. 0: The SDRAM does not enter power-down mode after being accessed. 1: The SDRAM enters power-down mode after being accessed.
8	BACTV	0	R/W	Bank Active Mode Specifies to access whether in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands) Note: Bank active mode can be set only in area 3, and only the 16-bit bus width can be set. When both the CS2 and CS3 spaces are set to SDRAM, specify auto-precharge mode.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4, 3	A3ROW[1:0]	00	R/W	Number of Bits of Row Address for Area 3 Specify the number of bits of the row address for area 3. 00: 11 bits 01: 12 bits 10: 13 bits 11: Reserved (setting prohibited)
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	A3COL[1:0]	00	R/W	Number of Bits of Column Address for Area 3 Specify the number of bits of the column address for area 3. 00: 8 bits 01: 9 bits 10: 10 bits 11: Reserved (setting prohibited)

8.4.5 Refresh Timer Control/Status Register (RTCSR)

RTCSR specifies various items about refresh for SDRAM. RTCSR is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

When RTCSR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection.

The phase of the clock for incrementing the count in the refresh timer counter (RTCNT) is adjusted only by a power-on reset. Note that there is an error in the time until the compare match flag is set for the first time after the timer is started with the CKS[2:0] bits being set to a value other than B'000.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7	CMF	0	R/W	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.

Bit	Bit Name	Initial Value	R/W	Description
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables CMF interrupt requests when the CMF bit in RTCSR is set to 1. 0: Disables CMF interrupt requests. 1: Enables CMF interrupt requests.
5 to 3	CKS[2:0]	000	R/W	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 000: Stop the counting-up 001: Bϕ/4 010: Bϕ/16 011: Bϕ/64 100: Bϕ/256 101: Bϕ/1024 110: Bϕ/2048 111: Bϕ/4096
2 to 0	RRC[2:0]	000	R/W	Refresh Count Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long. 000: 1 time 001: 2 times 010: 4 times 011: 6 times 100: 8 times 101: Reserved (setting prohibited) 110: Reserved (setting prohibited) 111: Reserved (setting prohibited)

8.4.6 Refresh Timer Counter (RTCNT)

RTCNT is an 8-bit counter that increments using the clock selected by bits CKS[2:0] in RTCSR. When RTCNT matches RTCOR, RTCNT is cleared to 0. The value in RTCNT returns to 0 after counting up to 255. When the RTCNT is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This counter is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Counter

8.4.7 Refresh Time Constant Register (RTCOR)

RTCOR is an 8-bit register. When RTCOR matches RTCNT, the CMF bit in RTCSR is set to 1 and RTCNT is cleared to 0.

When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored.

The $\overline{\text{REFOUT}}$ signal can be asserted when a refresh request is generated while the bus is released. For details, see the description of Relationship between Refresh Requests and Bus Cycles in section 8.5.6 (9), Relationship between Refresh Requests and Bus Cycles, and section 8.5.11, Bus Arbitration.

When the CMIE bit in RTCSR is set to 1, an interrupt request is issued by this matching signal. The request continues to be output until the CMF bit in RTCSR is cleared. Clearing the CMF bit only affects the interrupt request and does not clear the refresh request. Therefore, a combination of refresh request and interval timer interrupt can be specified so that the number of refresh requests are counted by using timer interrupts while refresh is performed periodically.

When RTCOR is written, the upper 16 bits of the write data must be H'A55A to cancel write protection. This register is initialized to H'00000000 by a power-on reset and retains the value by a manual reset and in software standby mode.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0.
7 to 0		All 0	R/W	8-Bit Register

8.5 Operation

8.5.1 Endian/Access Size and Data Alignment

This LSI supports big endian in which the 0 address is the most significant byte (MSB), and little endian in which the 0 address is the least significant byte (LSB) in the byte data. In a space of areas 1 to 7, endian can be set by the CSnBCR setting while the target space is not accessed. In a space of area 0, the CSnBCR setting is invalid in on-chip ROM-disabled mode. In on-chip ROM-enabled mode, endian can be set by the CSnBCR setting in a space of areas 0 to 7.

Two data bus widths (8 bits and 16 bits) are available for normal memory and SRAM with byte selection. Only 16-bit data bus width is available for SDRAM. 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 8.5 to 8.8 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. WE1 indicates the 0 address in big-endian mode, but WE0 indicates the 0 address in little-endian mode.

Table 8.5 16-Bit External Device Access and Data Alignment in Big-Endian Mode

		Data Bus		Strobe Signals	
Operation		**D15 to D8**	**D7 to D0**	**$\overline{\text{WE1}}$, $\overline{\text{DQMLU}}$**	**$\overline{\text{WE0}}$, $\overline{\text{DQMLL}}$**
Byte access at 0		Data 7 to 0	—	Assert	—
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		Data 7 to 0	—	Assert	—
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword access at 0	1st time at 0	Data 23 to 16	Data 31 to 24	Assert	Assert
	2nd time at 2	Data 7 to 0	Data 15 to 8	Assert	Assert

Table 8.6 8-Bit External Device Access and Data Alignment in Big-Endian Mode

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, $\overline{DQMLU}$	$\overline{WE0}$, $\overline{DQMLL}$
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0	1st time at 0	—	Data 15 to 8	—	Assert
	2nd time at 1	—	Data 7 to 0	—	Assert
Word access at 2	1st time at 2	—	Data 15 to 8	—	Assert
	2nd time at 3	—	Data 7 to 0	—	Assert
Longword access at 0	1st time at 0	—	Data 31 to 24	—	Assert
	2nd time at 2	—	Data 23 to 16	—	Assert
	3rd time at 2	—	Data 15 to 8	—	Assert
	4th time at 3	—	Data 7 to 0	—	Assert

Table 8.7 16-Bit External Device Access and Data Alignment in Little-Endian Mode

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, $\overline{DQMLU}$	$\overline{WE0}$, $\overline{DQMLL}$
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		Data 7 to 0	—	Assert	—
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		Data 7 to 0	—	Assert	—
Word access at 0		Data 15 to 8	Data 7 to 0	Assert	Assert
Word access at 2		Data 15 to 8	Data 7 to 0	Assert	Assert
Longword access at 0	1st time at 0	Data 15 to 8	Data 7 to 0	Assert	Assert
	2nd time at 2	Data 31 to 24	Data 23 to 16	Assert	Assert

Table 8.8 8-Bit External Device Access and Data Alignment in Little-Endian Mode

Operation		Data Bus		Strobe Signals	
		D15 to D8	D7 to D0	$\overline{WE1}$, $\overline{DQMLU}$	$\overline{WE0}$, $\overline{DQMLL}$
Byte access at 0		—	Data 7 to 0	—	Assert
Byte access at 1		—	Data 7 to 0	—	Assert
Byte access at 2		—	Data 7 to 0	—	Assert
Byte access at 3		—	Data 7 to 0	—	Assert
Word access at 0	1st time at 0	—	Data 7 to 0	—	Assert
	2nd time at 1	—	Data 15 to 8	—	Assert
Word access at 2	1st time at 2	—	Data 7 to 0	—	Assert
	2nd time at 3	—	Data 15 to 8	—	Assert
Longword access at 0	1st time at 0	—	Data 7 to 0	—	Assert
	2nd time at 2	—	Data 15 to 8	—	Assert
	3rd time at 2	—	Data 23 to 16	—	Assert
	4th time at 3	—	Data 31 to 24	—	Assert

8.5.2 Normal Space Interface

(1) Basic Timing

For access to a normal space, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 8.5.8, SRAM Interface with Byte Selection. Figure 8.2 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.

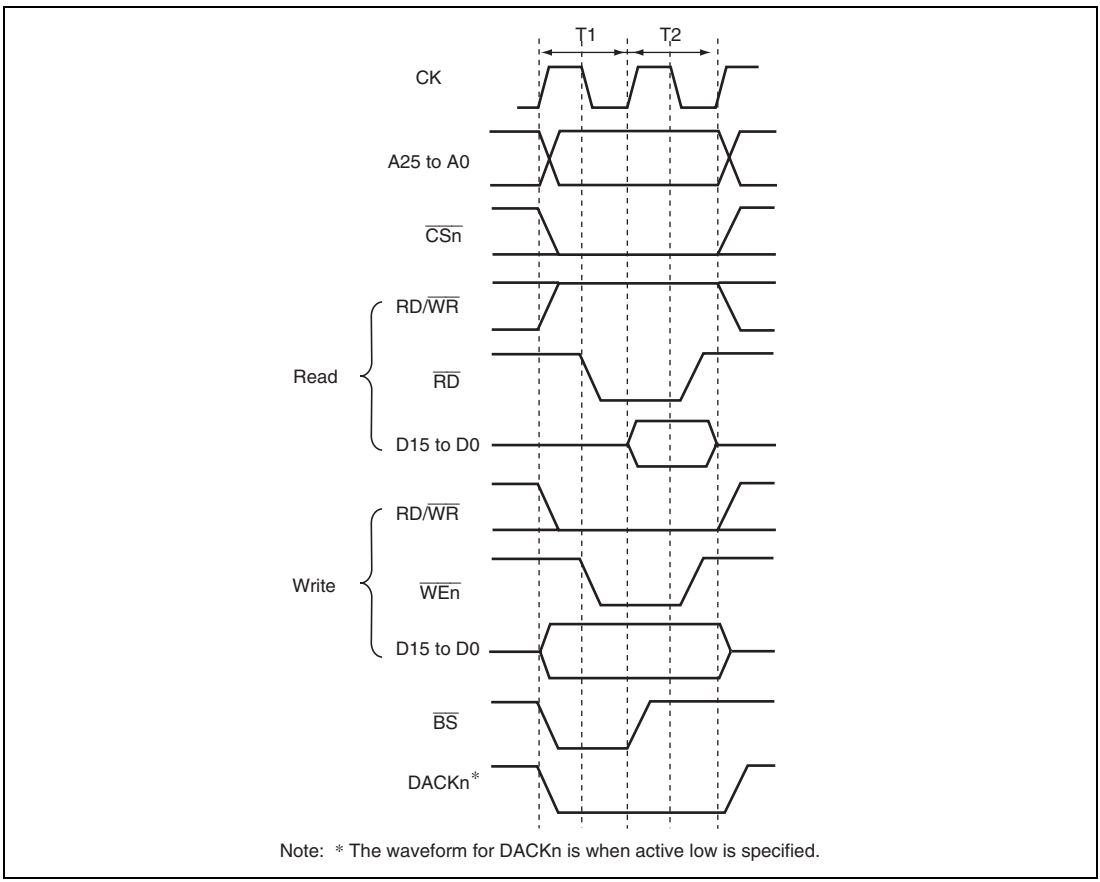


Figure 8.2 Normal Space Basic Access Timing (Access Wait 0)

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 $\overline{WEn}$ signal for the byte to be written is asserted.

It is necessary to output the data that has been read using $\overline{RD}$ when a buffer is established in the data bus. The RD/$\overline{WR}$ signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer, to avoid collision.

Figures 8.3 and 8.4 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 8.3). If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (figure 8.4).

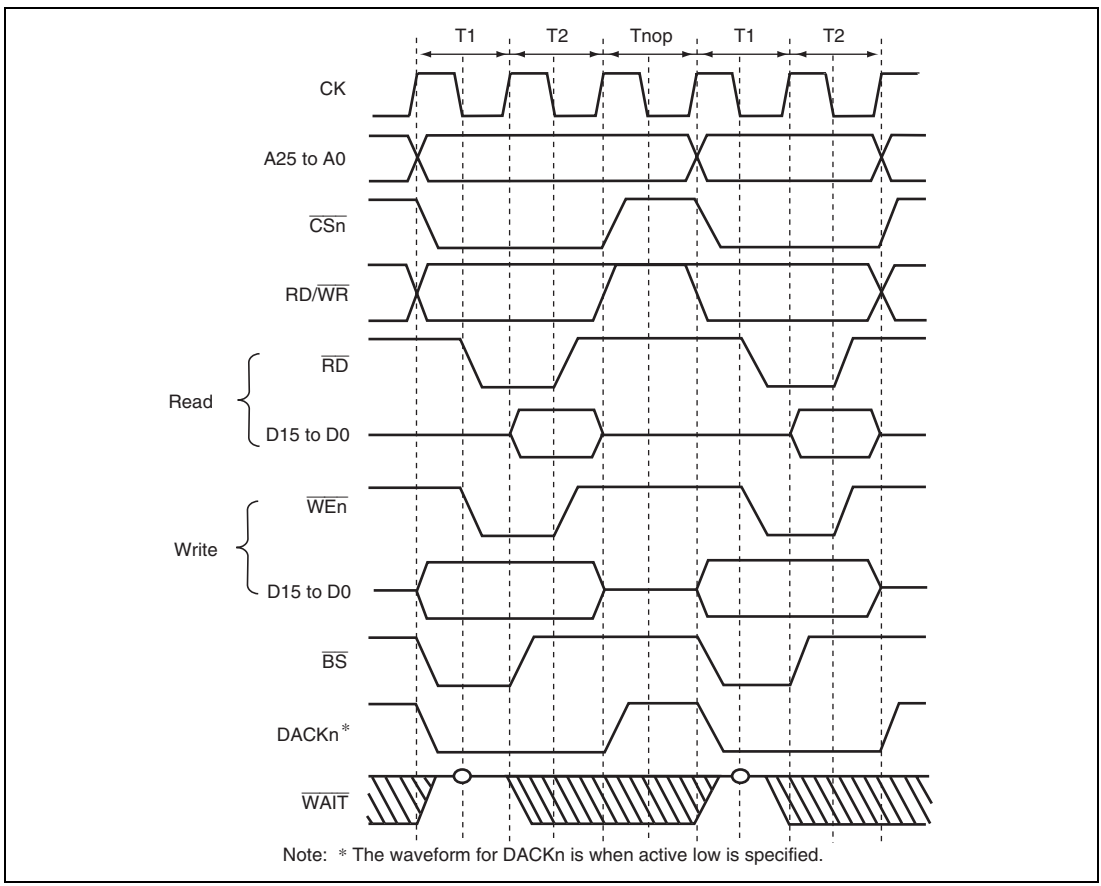


Figure 8.3 Continuous Access for Normal Space 1
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 0
(Access Wait = 0, Cycle Wait = 0)

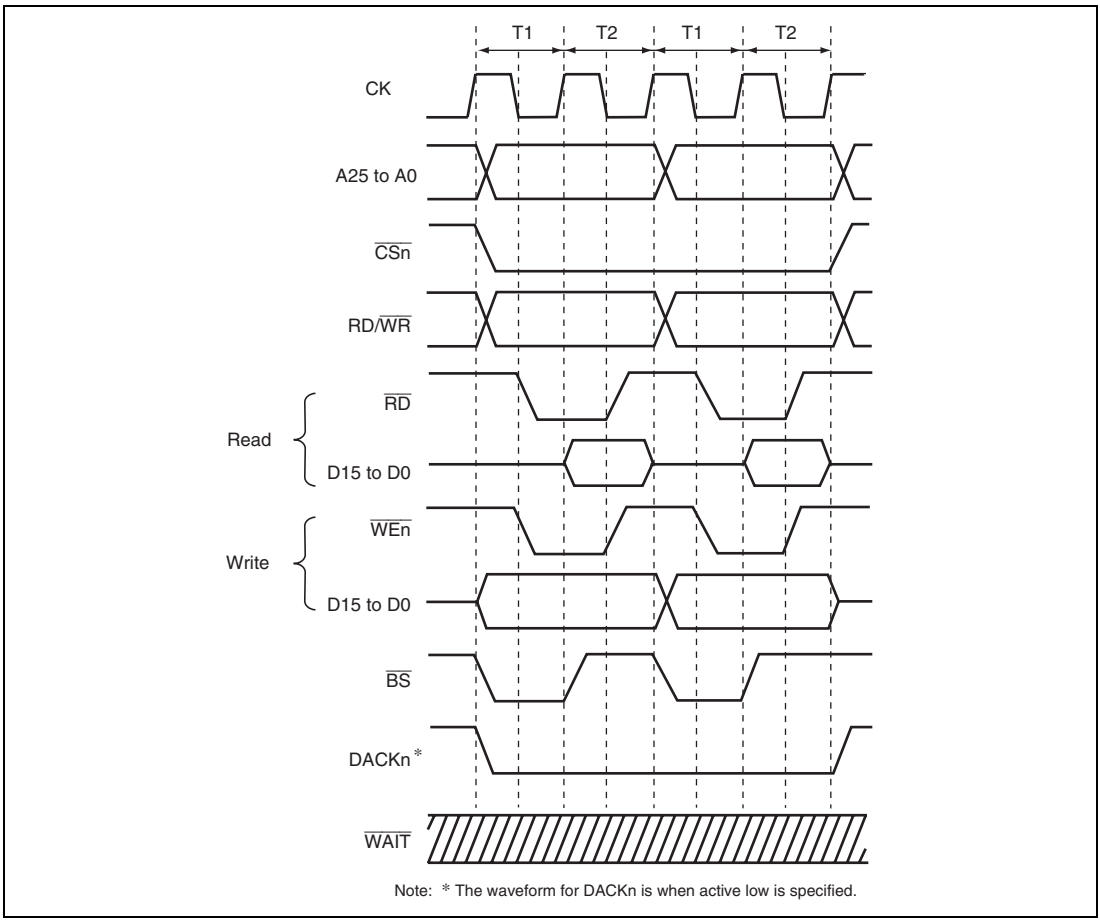


Figure 8.4 Continuous Access for Normal Space 2
Bus Width = 16 Bits, Longword Access, CSnWCR.WM Bit = 1
(Access Wait = 0, Cycle Wait = 0)

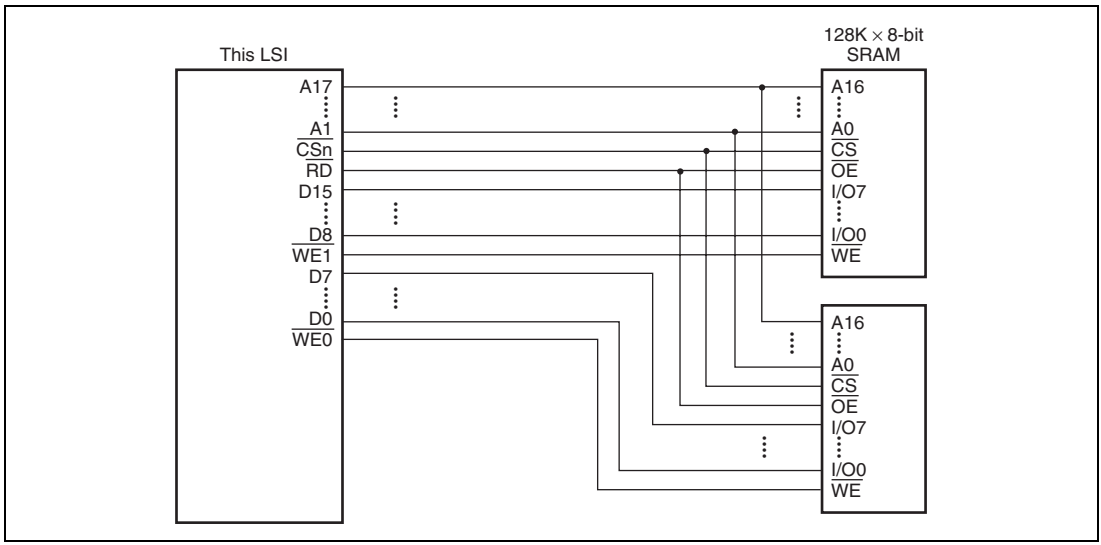


Figure 8.5 Example of 16-Bit Data-Width SRAM Connection

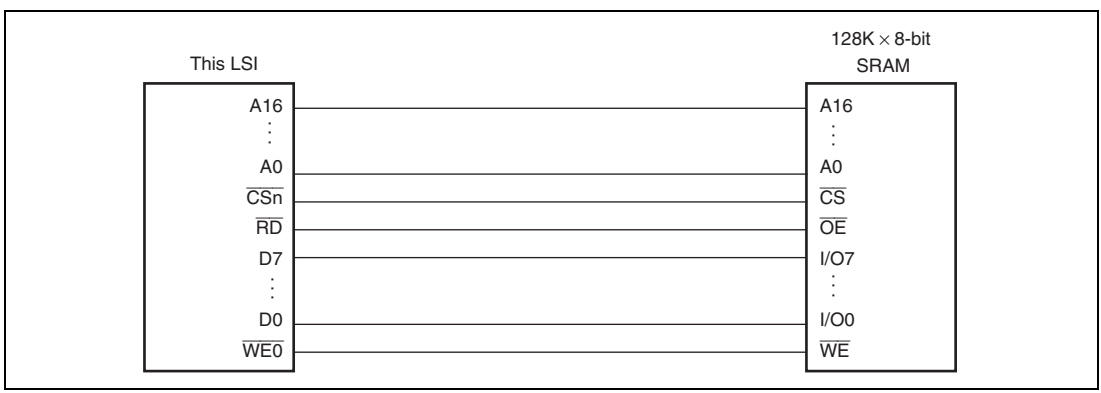
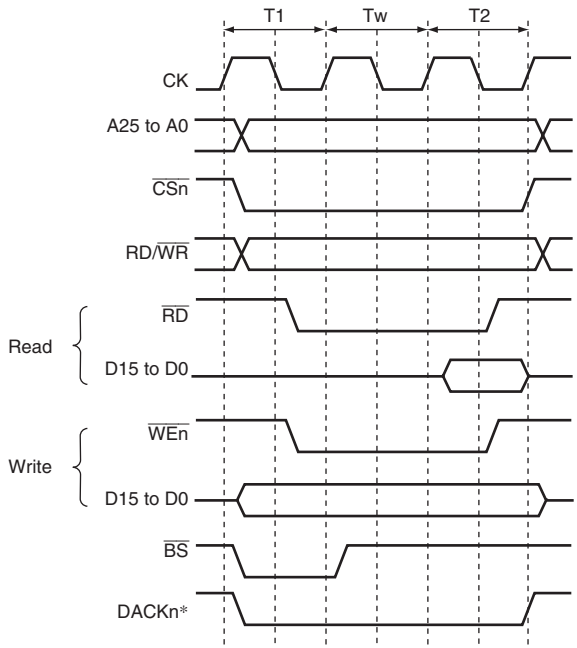


Figure 8.6 Example of 8-Bit Data-Width SRAM Connection

8.5.3 Access Wait Control

Wait cycle insertion on a normal space access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for areas 1, 4, 5, and 7 to insert wait cycles independently in read access and in write access. Areas 0, 2, 3, and 6 have common access wait for read cycle and write cycle. The specified number of Tw cycles are inserted as wait cycles in a normal space access shown in figure 8.7.



Note: * The waveform for DACKn is when active low is specified.

Figure 8.7 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 8.8. 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.

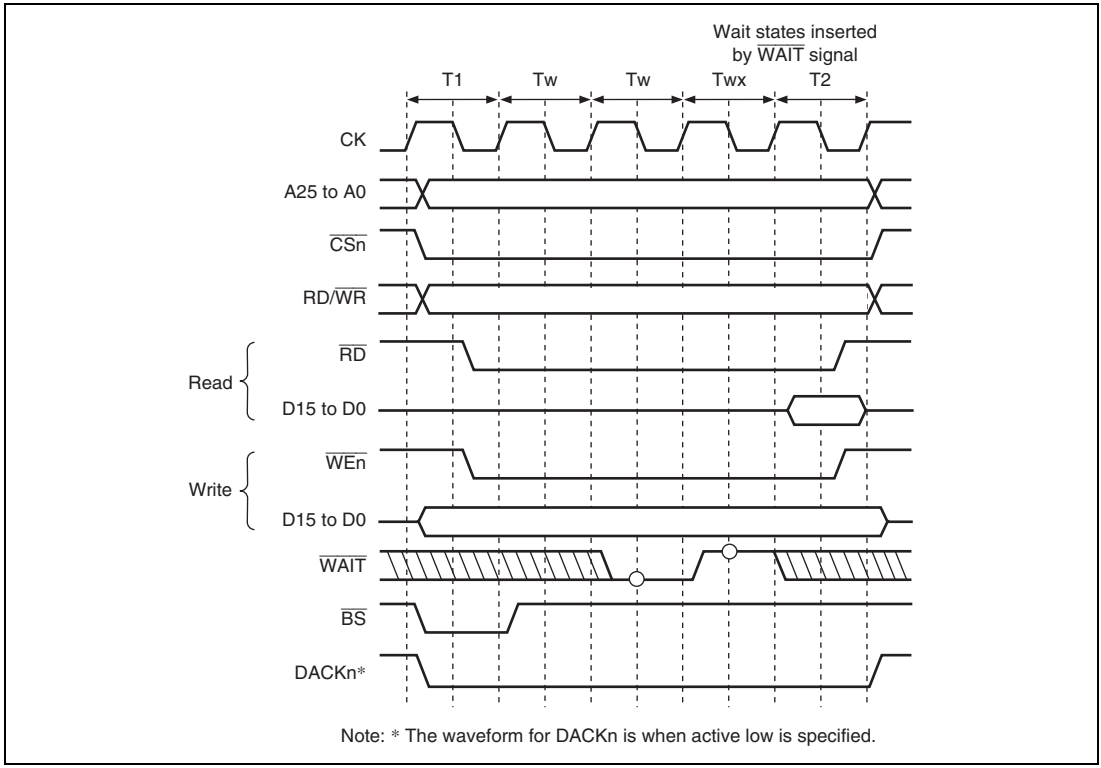


Figure 8.8 Wait Cycle Timing for Normal Space Access (Wait Cycle Insertion Using $\overline{\text{WAIT}}$ Signal)

8.5.4 $\overline{\text{CSn}}$ Assert Period Expansion

The number of cycles from $\overline{\text{CSn}}$ assertion to $\overline{\text{RD}}$, $\overline{\text{WEn}}$ assertion can be specified by setting bits SW1 and SW0 in CSnWCR. The number of cycles from $\overline{\text{RD}}$, $\overline{\text{WEn}}$ 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 8.9 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 $\overline{\text{WEn}}$ 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.

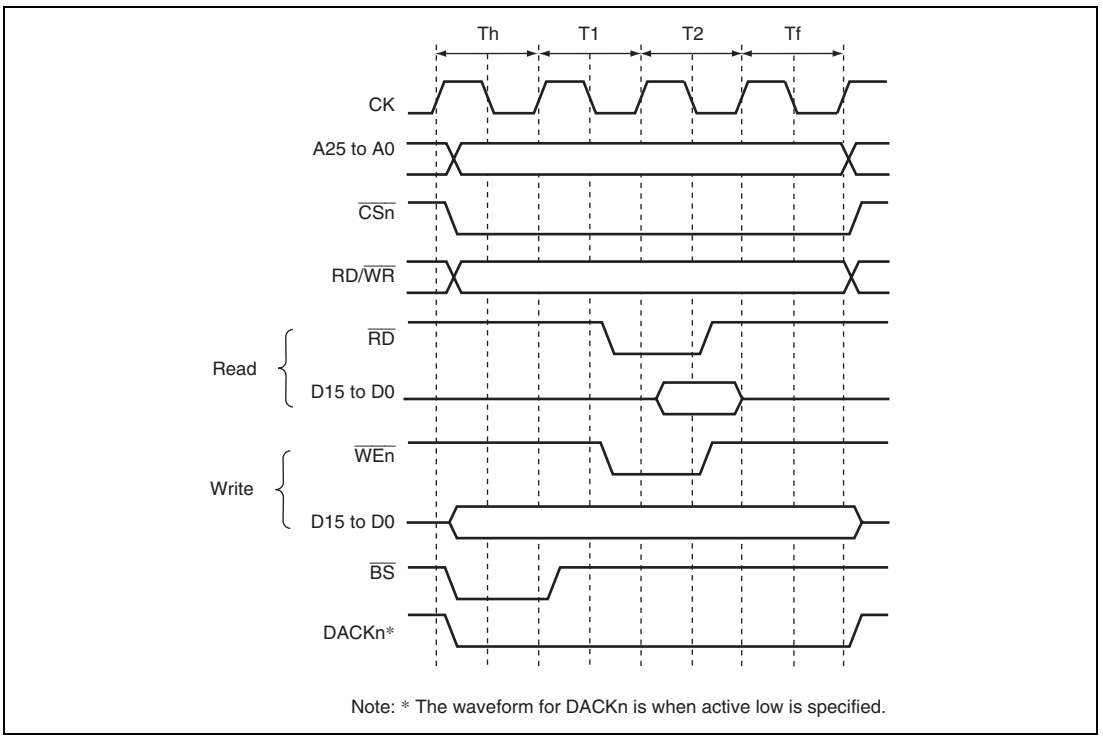


Figure 8.9 $\overline{\text{CSn}}$ Assert Period Expansion

8.5.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, $\overline{CS5}$, $\overline{AH}$, $\overline{RD}$, and $\overline{WEn}$ signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to a normal space. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits. Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from cycle Ta2 to cycle Ta3. Because cycle Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in CS5WCR to 1.

The RD/$\overline{WR}$ signal is output at the same time as the $\overline{CS5}$ signal; it is high in the read cycle and low in the write cycle.

The data cycle is the same as that in a normal space access.

Timing charts are shown in figures 8.10 to 8.12.

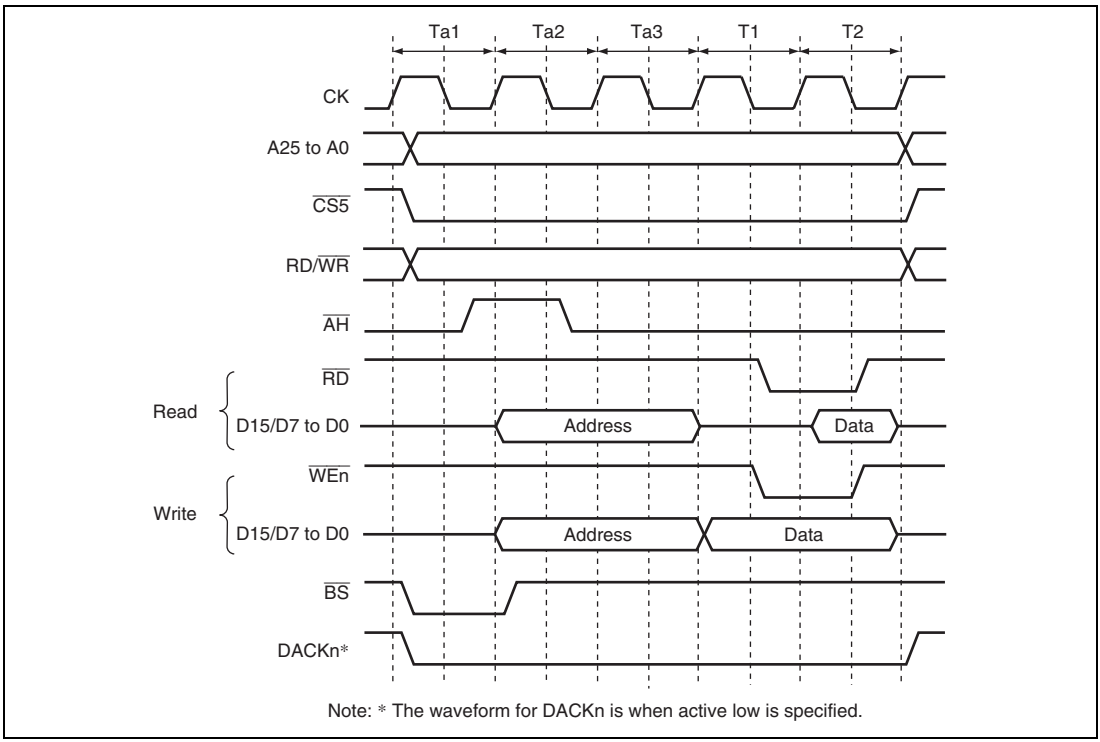


Figure 8.10 Access Timing for MPX Space (Address Cycle No Wait, Data Cycle No Wait)

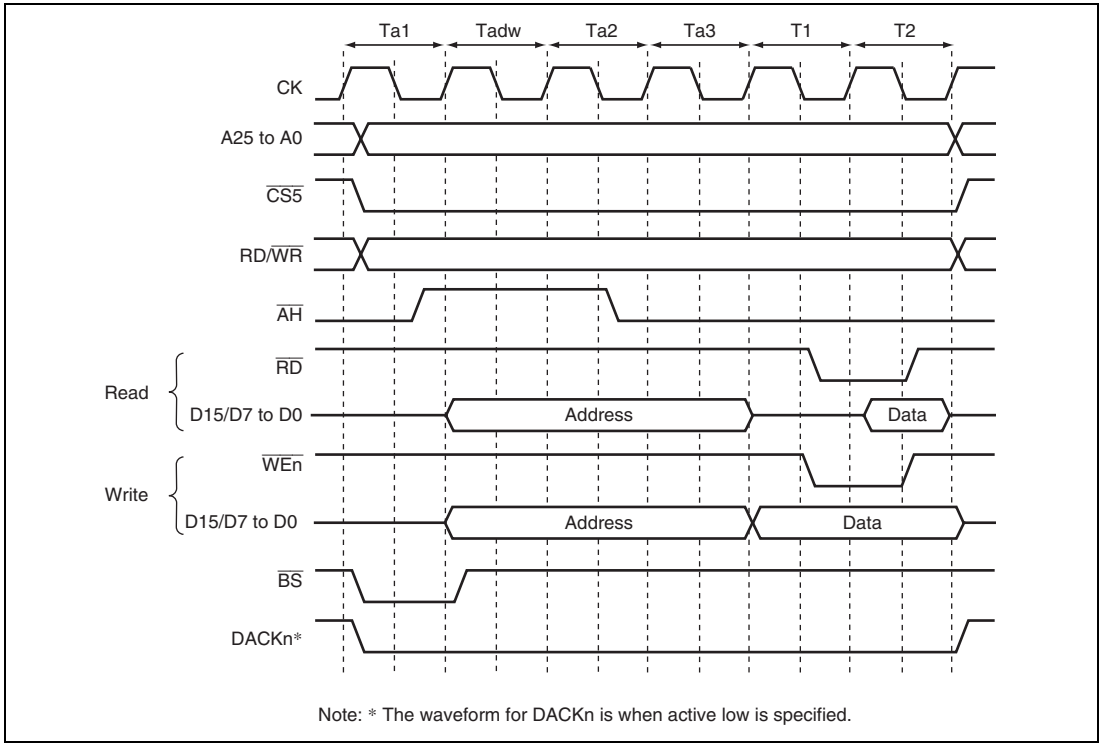


Figure 8.11 Access Timing for MPX Space (Address Cycle Wait 1, Data Cycle No Wait)

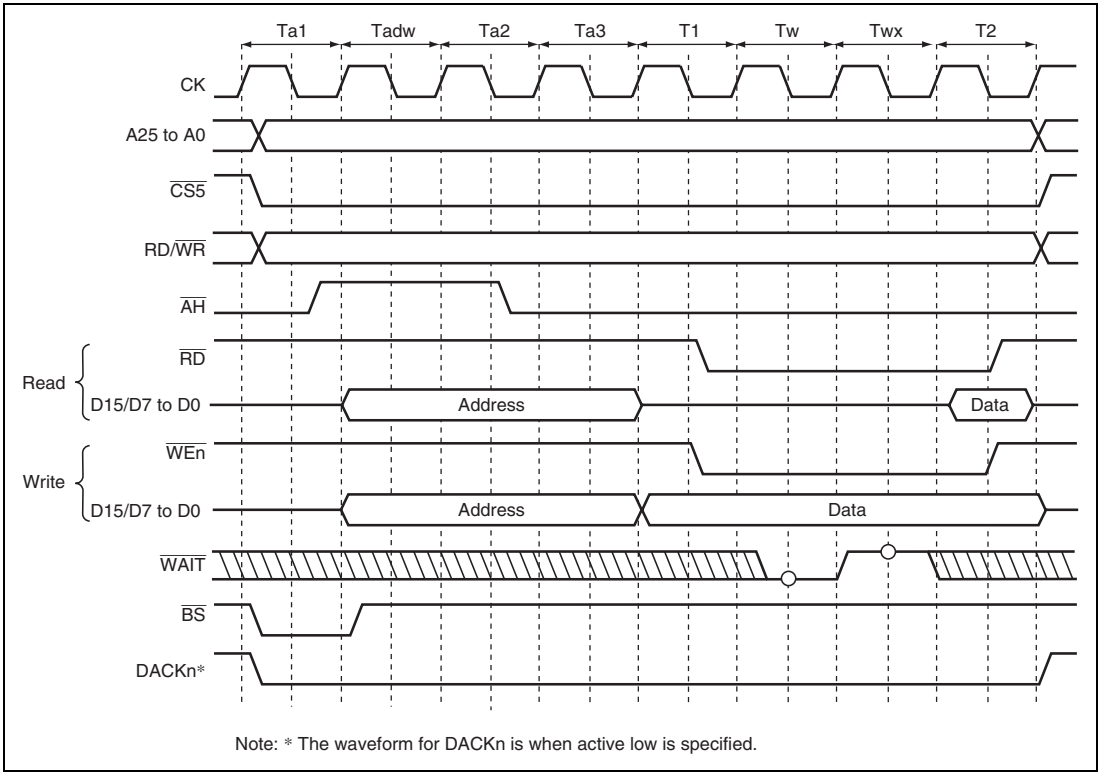


Figure 8.12 Access Timing for MPX Space (Address Cycle Access Wait 1, Data Cycle Wait 1, External Wait 1)

8.5.6 SDRAM Interface

(1) SDRAM Direct Connection

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.

The control signals for direct connection of SDRAM are $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, $\overline{\text{DQMUL}}$, $\overline{\text{DQMLL}}$, CKE, $\overline{\text{CS2}}$, and $\overline{\text{CS3}}$. All the signals other than $\overline{\text{CS2}}$ and $\overline{\text{CS3}}$ are common to all areas, and signals other than CKE are valid when $\overline{\text{CS2}}$ or $\overline{\text{CS3}}$ is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM can be set to 16 bits only.

Burst read/single write (burst length 1) and burst read/burst write (burst length 1) are supported as SDRAM operating mode.

Commands for SDRAM can be specified by $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, RD/$\overline{\text{WR}}$, and specific address signals. These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)
- Write (WRIT)
- Write with pre-charge (WRITA)
- Write mode register (MRS, EMRS)

The byte to be accessed is specified by $\overline{\text{DQMUL}}$ and $\overline{\text{DQMLL}}$. Reading or writing is performed for a byte whose corresponding $\overline{\text{DQMxx}}$ is low. For details on the relationship between $\overline{\text{DQMxx}}$ and the byte to be accessed, see section 8.5.1, Endian/Access Size and Data Alignment.

Figure 8.13 shows an example of the connection of the SDRAM with the LSI.

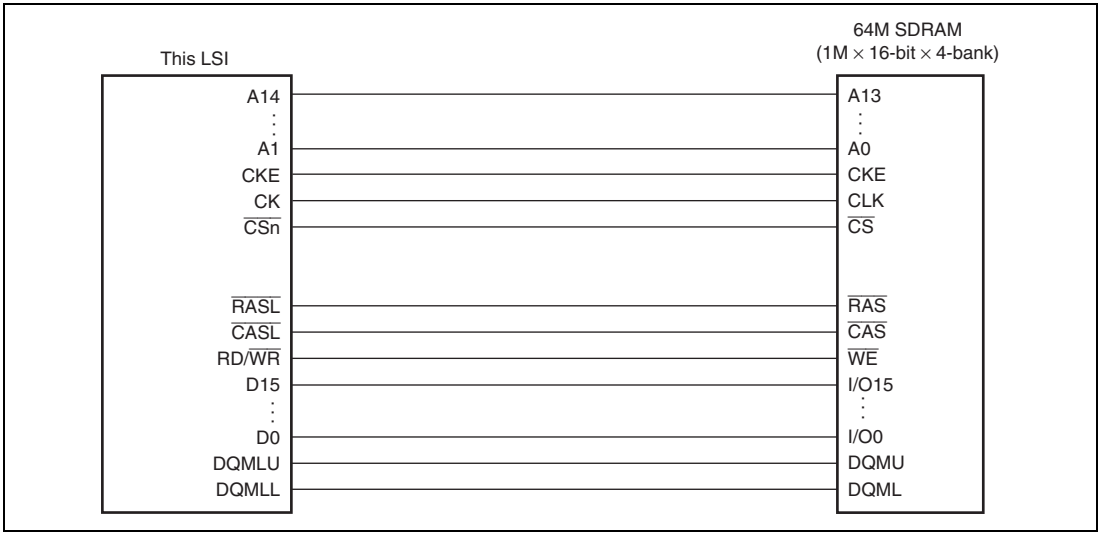


Figure 8.13 Example of 16-Bit Data Width SDRAM Connection ($\overline{\text{RASU}}$ and $\overline{\text{CASU}}$ are Not Used)

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR, bits A2ROW[1:0], and A2COL[1:0], A3ROW[1:0], and A3COL[1:0] in SDCR. Tables 8.9 to 8.11 show the relationship between the settings of bits BSZ[1:0], A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A29 to A18 are not multiplexed and the original values of address are always output at these pins.

The A0 pin of SDRAM specifies a word address. Therefore, connect the A0 pin of SDRAM to the A1 pin of the LSI; then connect the A1 pin of SDRAM to the A2 pin of the LSI, and so on.

Table 8.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*[2]	A21*[2]	A12 (BA1)	Specifies bank
A12	A20*[2]	A20*[2]	A11 (BA0)	
A11	A19	L/H*[1]	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

Table 8.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (1)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Specifies bank
A13	A21*2	A21*2	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mword × 16 bits × 4 banks, column 8 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at L or H according to access the mode.
2. Bank address specification

Table 8.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies bank
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 8.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (2)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13 (BA1)	Specifies bank
A13	A23*2	A23*2	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 8.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)-1

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*[2]	A24*[2]	A14 (BA1)	Specifies bank
A14	A23*[2]	A23*[2]	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*[1]	A10/AP	Specifies address/precharge
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.

2. Bank address specification

Table 8.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)-2

Setting				
BSZ [1:0]	**A2/3 ROW [1:0]**	**A2/3 COL [1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output Pin of This LSI	**Row Address Output Cycle**	**Column Address Output Cycle**	**SDRAM Pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2	A25*2	A14 (BA1)	Specifies bank
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/precharge
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Notes: 1. L/H is a bit used in the command specification; it is fixed at low or high according to the access mode.
2. Bank address specification

(3) Burst Read

A burst read occurs in the following cases with this LSI.

- Access size in reading is larger than data bus width.
- 16-byte transfer in DMAC

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 8 times to read 16-byte continuous data from the SDRAM that is connected to a 16-bit data bus. This access is called the burst read with the burst number 8. Table 8.12 shows the relationship between the access size and the number of bursts.

Table 8.12 Relationship between Access Size and Number of Bursts

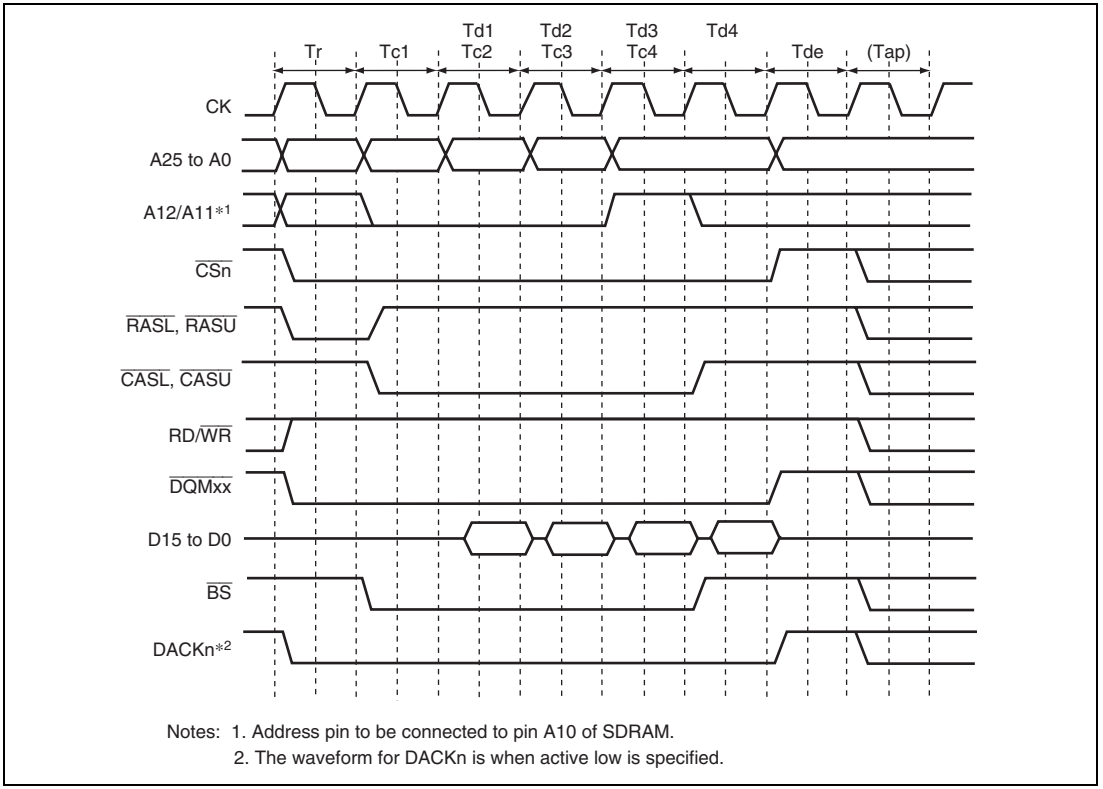
Bus Width	Access Size	Number of Bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bits	8

Figures 8.14 and 8.15 show a timing chart in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CK) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.

In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect the SDRAM in variable frequencies. Figure 8.15 shows an example in which wait cycles are inserted. The number of cycles from the Tr cycle where the ACTV command is output to the Tc1 cycle where the READ command is output can be specified using the WTRCD1 and WTRCD0 bits in CS3WCR. If the WTRCD1 and WTRCD0 bits specify one cycles or more, a Trw cycle where the NOT command is issued is inserted between the Tr cycle and Tc1 cycle. The number of cycles from the Tc1 cycle where the READ command is output to the Td1 cycle where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in CS2WCR or the A3CL1 and A3CL0 bits in CS3WCR and WTRCD0 bit in CS3WCR. The number of cycles from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS latency for the SDRAM is normally defined as up to three cycles. However, the CAS latency in this LSI can

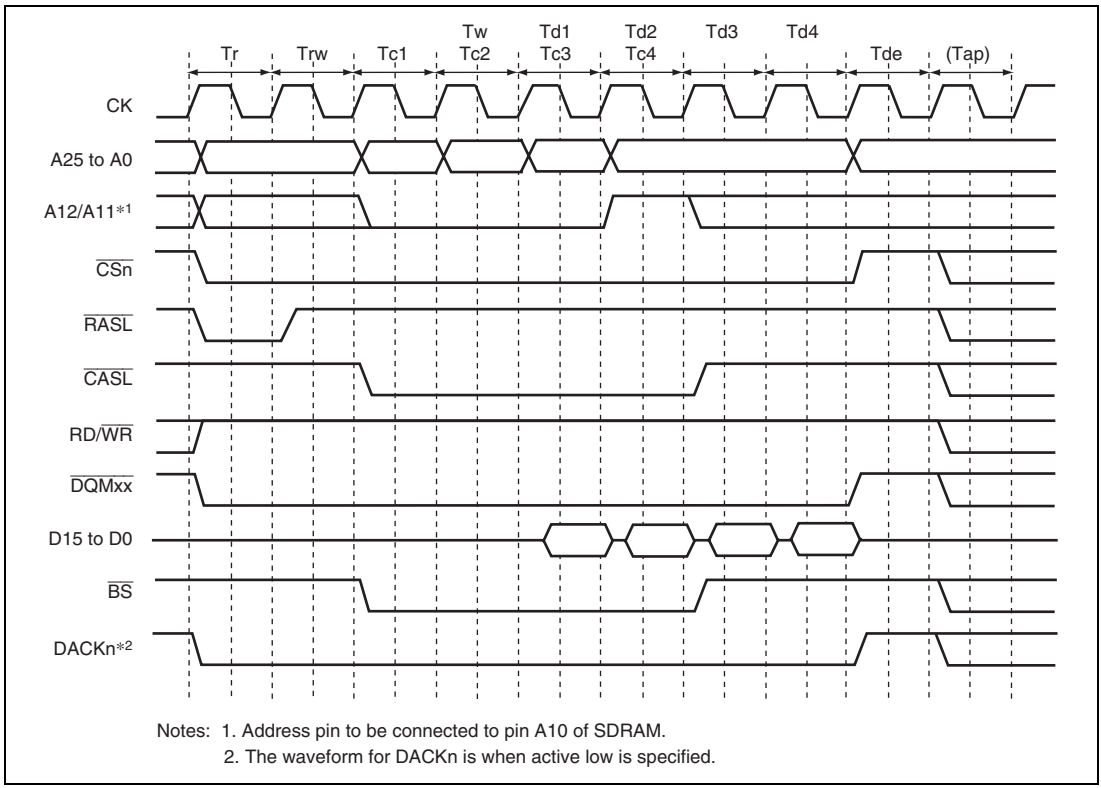
be specified as 1 to 4 cycles. This CAS latency can be achieved by connecting a latch circuit between this LSI and the SDRAM.

A Tde cycle is an idle cycle required to transfer the read data into this LSI and occurs once for every burst read or every single read.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.14 Burst Read Basic Timing (CAS Latency 1, Auto-Precharge)



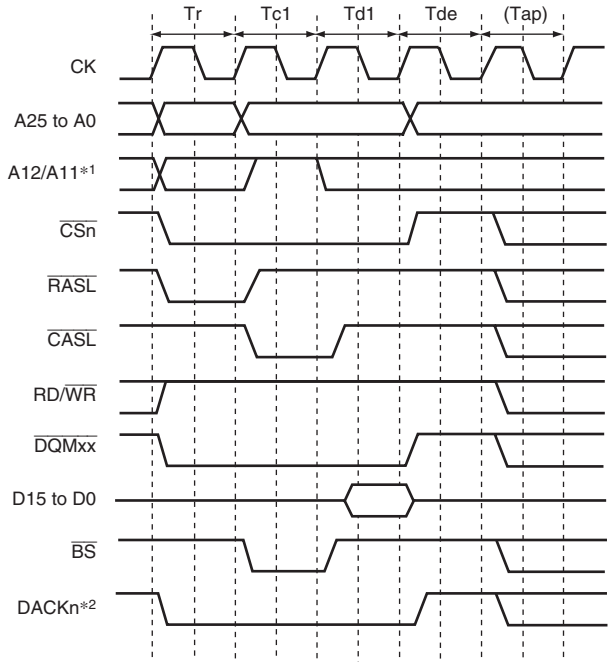
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.15 Burst Read Wait Specification Timing (CAS Latency 2, WTRCD[1:0] = 1 Cycle, Auto-Precharge)

(4) Single Read

A read access ends in one cycle when the data bus width is larger than or equal to the access size. This, simply stated, is single read. As the SDRAM is set to the burst read with the burst length 1, only the required data is output. A read access that ends in one cycle is called single read.

Figure 8.16 shows the single read basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.16 Basic Timing for Single Read (CAS Latency 1, Auto-Precharge)

(5) Burst Write

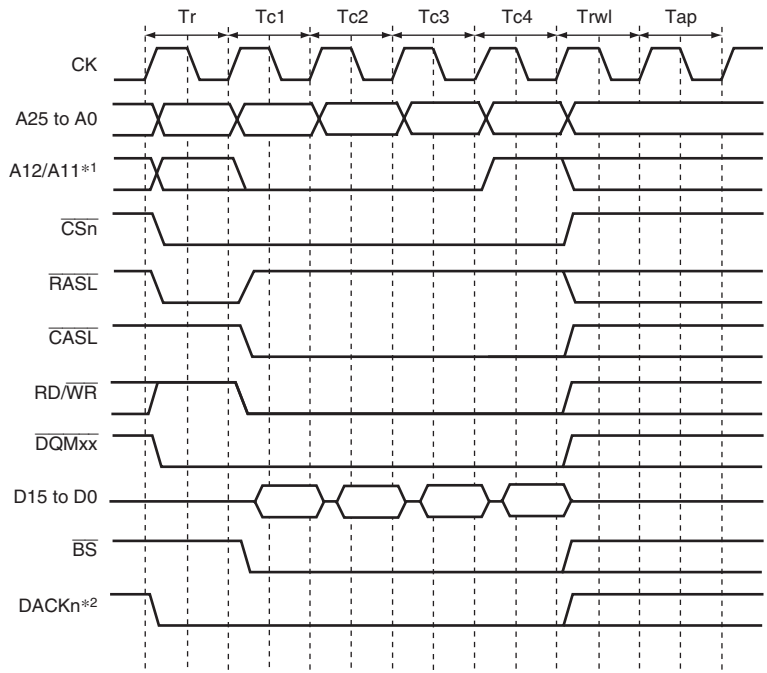
A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- 16-byte transfer in DMAC

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 8 times to write 16-byte continuous data to the SDRAM that is connected to a 16-bit data bus. This access is called burst write with the burst number 8.

The relationship between the access size and the number of bursts is shown in table 8.12.

Figure 8.17 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr cycle, the WRIT command is issued in the Tc1, Tc2, and Tc3 cycles, and the WRITA command is issued to execute an auto-precharge in the Tc4 cycle. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 cycle that waits for the auto-precharge initiation is followed by the Tap cycle that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. Between the Trwl and the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 cycles is specified by the TRWL1 and TRWL0 bits in CS3WCR. The number of Tap cycles is specified by the WTRP1 and WTRP0 bits in CS3WCR.

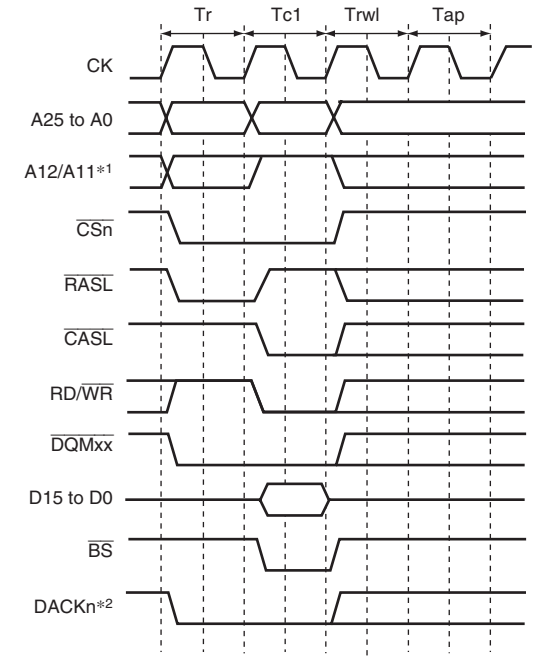


Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.17 Basic Timing for Burst Write (Auto-Precharge)

(6) Single Write

A write access ends in one cycle when the data bus width is larger than or equal to access size. As a single write or burst write with burst length 1 is set in SDRAM, only the required data is output. The write access that ends in one cycle is called single write. Figure 8.18 shows the single write basic timing.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.18 Single Write Basic Timing (Auto-Precharge)

(7) Bank Active

The SDRAM bank function can be used to support high-speed access to the same row address. When the BACTV bit in SDCR is 1, access is performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for either the upper or lower bits of area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or SRAM with byte selection. When areas 2 and 3 are both set to SDRAM or both the upper and lower bits of area 3 are connected to SDRAM, auto-precharge mode must be set.

When the bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As SDRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is issued. The number of cycles between issuance of the PRE command and the ACTV command is determined by the WTRP1 and WTPR0 bits in CS3WCR.

In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap cycles after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap cycles for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS.

A burst read cycle without auto-precharge is shown in figure 8.19, a burst read cycle for the same row address in figure 8.20, and a burst read cycle for different row addresses in figure 8.21. Similarly, a burst write cycle without auto-precharge is shown in figure 8.22, a burst write cycle for the same row address in figure 8.23, and a burst write cycle for different row addresses in figure 8.24.

In figure 8.20, a Tnop cycle in which no operation is performed is inserted before the Tc cycle that issues the READ command. The Tnop cycle is inserted to acquire two cycles of CAS latency for the $\overline{\text{DQMxx}}$ signal that specifies the read byte in the data read from the SDRAM. If the CAS

latency is specified as two cycles or more, the Tnop cycle is not inserted because the two cycles of latency can be acquired even if the $\overline{\text{DQMxx}}$ signal is asserted after the Tc cycle.

When bank active mode is set, if only access cycles to the respective banks in the area 3 space are considered, as long as access cycles to the same row address continue, the operation starts with the cycle in figure 8.19 or 8.23, followed by repetition of the cycle in figure 8.20 or 8.23. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, after this is detected the bus cycle in figure 8.20 or 8.23 is executed instead of that in figure 8.21 or 8.24. In bank active mode, too, all banks become inactive after a refresh cycle or after the bus is released as the result of bus arbitration.

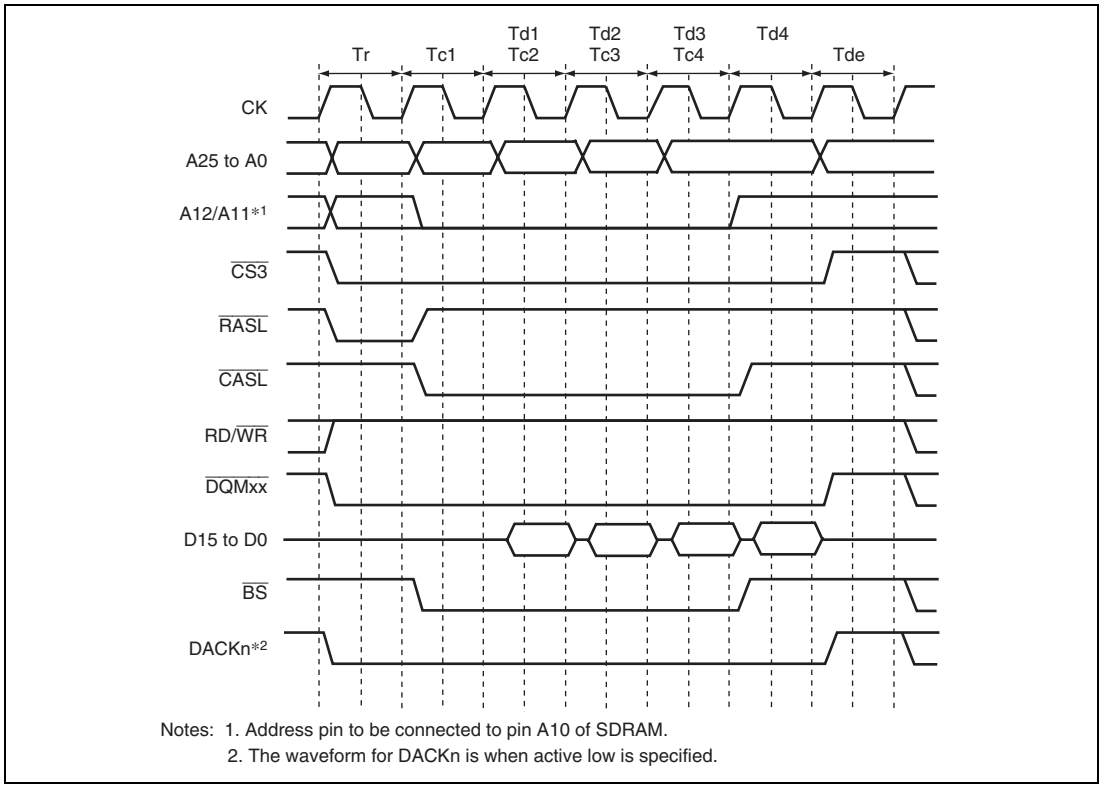


Figure 8.19 Burst Read Timing (Bank Active, Different Bank, CAS Latency 1)

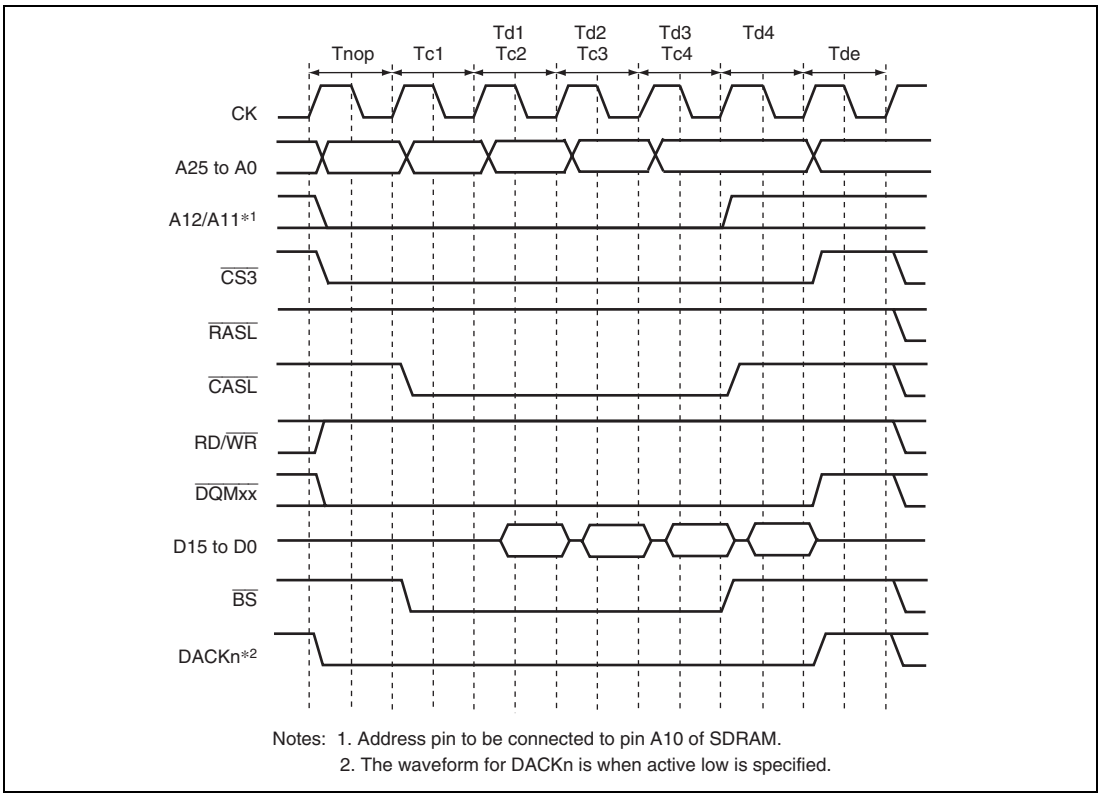
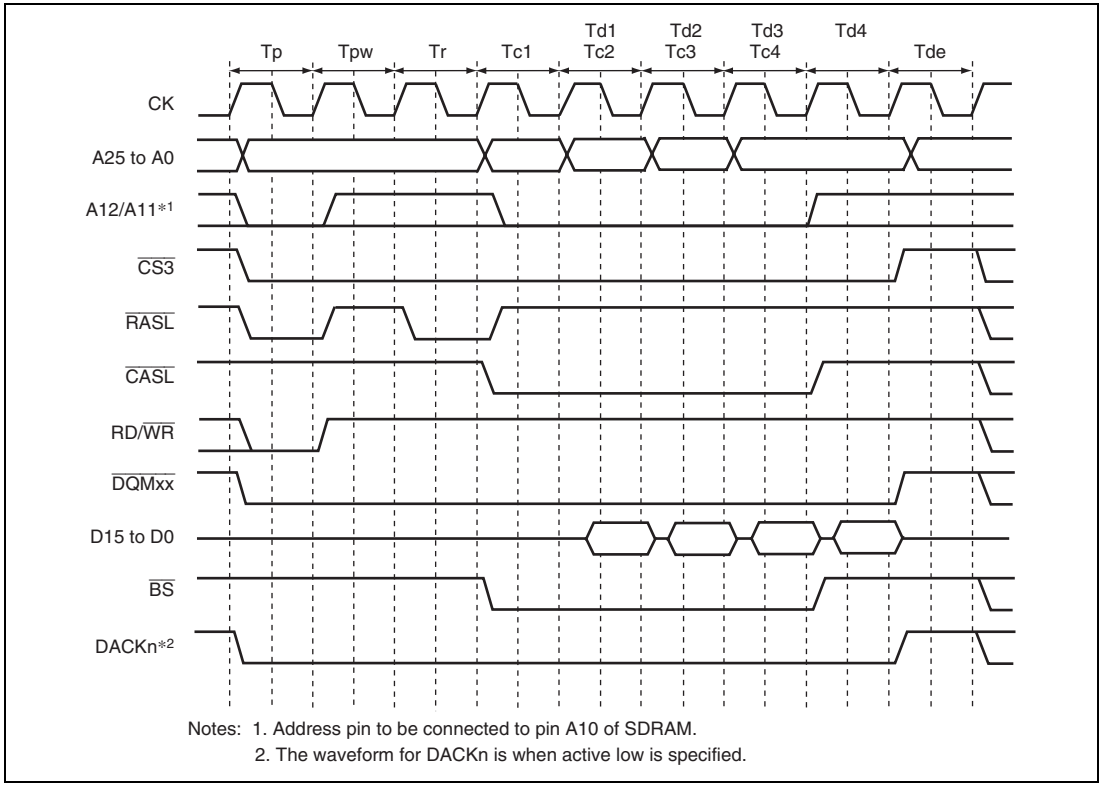
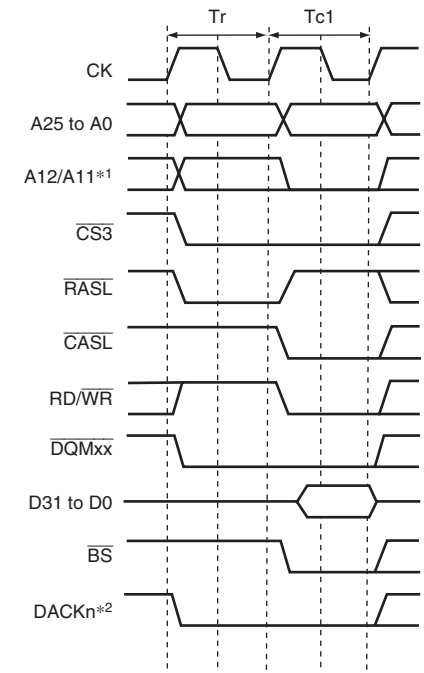


Figure 8.20 Burst Read Timing (Bank Active, Same Row Addresses in the Same Bank, CAS Latency 1)



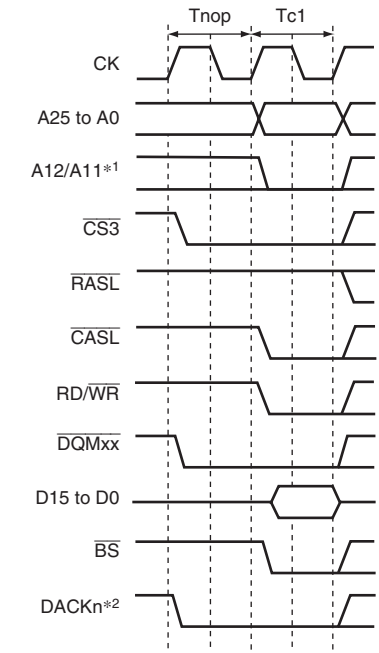
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.21 Burst Read Timing (Bank Active, Different Row Addresses in the Same Bank, CAS Latency 1)



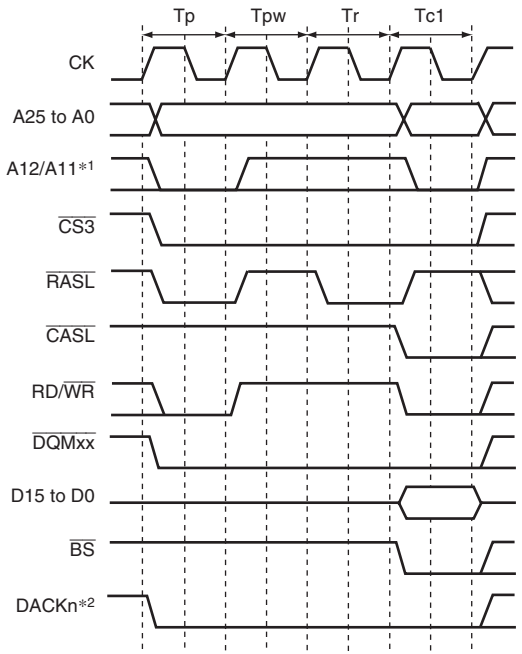
Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.22 Single Write Timing (Bank Active, Different Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.23 Single Write Timing (Bank Active, Same Row Addresses in the Same Bank)



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.24 Single Write Timing (Bank Active, Different Row Addresses in the Same Bank)

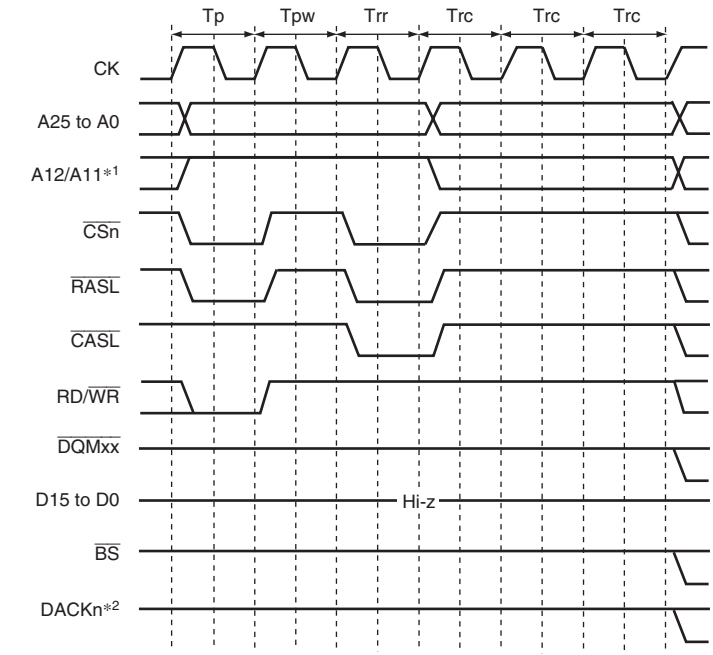
(8) Refreshing

This LSI has a function for controlling SDRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC2 to RRC0 bits in RTCSR. If SDRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

(a) Auto-refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, then make the CKS2 to CKS0 and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC2 to RRC0. At the same time, RTCNT is cleared to zero and the count-up is restarted.

Figure 8.25 shows the auto-refresh cycle timing. After starting, the auto refreshing, PALL command is issued in the Tp cycle to make all the banks to pre-charged state from active state when some bank is being pre-charged. Then REF command is issued in the Trr cycle after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR. A new command is not issued for the duration of the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR after the Trr cycle. The WTRC1 and WTRC0 bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle cycle is inserted between the Tp cycle and Trr cycle when the setting value of the WTRP1 and WTRP0 bits in CS3WCR is longer than or equal to 1 cycle.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.25 Auto-Refresh Timing

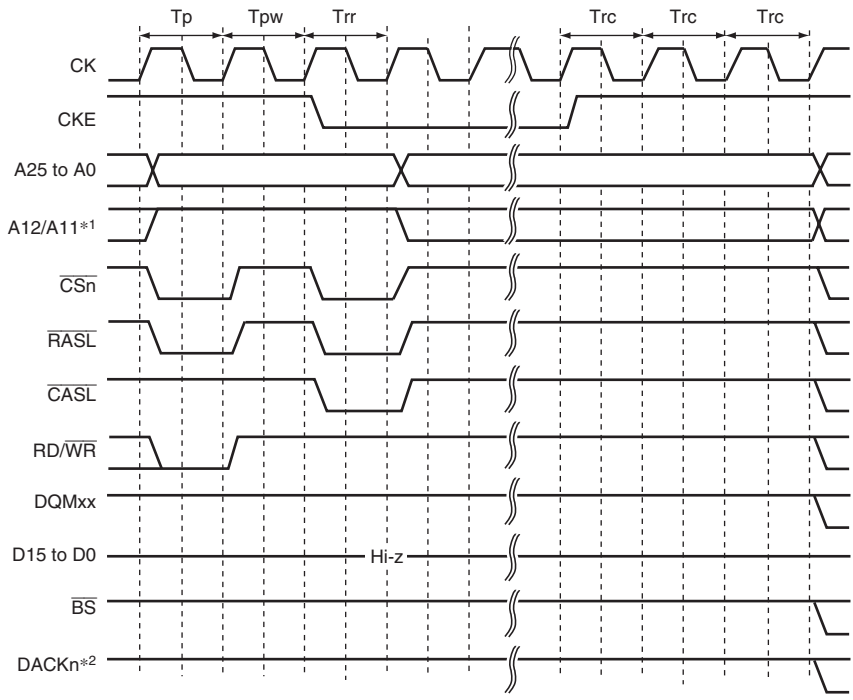
(b) Self-refreshing

Self-refresh mode in which the refresh timing and refresh addresses are generated within the SDRAM. Self-refreshing is activated by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp cycle after the completion of the pre-charging bank. A SELF command is then issued after inserting idle cycles of which number is specified by the WTRP1 and WTRP0 bits in CS3WSR. SDRAM cannot be accessed while in the self-refresh state. Self-refresh mode is cleared by clearing the RMODE bit to 0. After self-refresh mode has been cleared, command issuance is disabled for the number of cycles specified by the WTRC1 and WTRC0 bits in CS3WCR.

Self-refresh timing is shown in figure 8.26. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, or when exiting standby mode other than through a power-on reset, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using the LSI standby function, and is maintained even after recovery from standby mode due to an interrupt. Note that the necessary signals such as CKE must be driven even in standby state by setting the HIZCNT bit in CMNCR to 1.

The self-refresh state is not cleared by a manual reset. In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Notes: 1. Address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn is when active low is specified.

Figure 8.26 Self-Refresh Timing

(9) Relationship between Refresh Requests and Bus Cycles

If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed. If a refresh request occurs while the bus is released by the bus arbitration function, the refresh will not be executed until the bus mastership is acquired. This LSI has the $\overline{\text{REFOUT}}$ pin to request the bus while waiting for refresh execution. For $\overline{\text{REFOUT}}$ pin function selection, see section 19, Pin Function Controller (PFC). This LSI continues to assert $\overline{\text{REFOUT}}$ (low level) until the bus is acquired.

On receiving the asserted $\overline{\text{REFOUT}}$ signal, the external device must negate the $\overline{\text{BREQ}}$ signal and return the bus. If the external bus does not return the bus for a period longer than the specified refresh interval, refresh cannot be executed and the SDRAM contents may be lost.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval or the bus mastership occupation must be prevented from occurring.

If a bus mastership is requested during self-refresh, the bus will not be released until the refresh is completed.

(10) Low-Frequency Mode

When the SLOW bit in SDCR is set to 1, output of commands, addresses, and write data, and fetch of read data are performed at a timing suitable for operating SDRAM at a low frequency.

Figure 8.27 shows the access timing in low-frequency mode. In this mode, commands, addresses, and write data are output in synchronization with the falling edge of CK, which is half a cycle delayed than the normal timing. Read data is fetched at the rising edge of CK, which is half a cycle faster than the normal timing. This timing allows the hold time of commands, addresses, write data, and read data to be extended.

If SDRAM is operated at a high frequency with the SLOW bit set to 1, the setup time of commands, addresses, write data, and read data are not guaranteed. Take the operating frequency and timing design into consideration when making the SLOW bit setting.

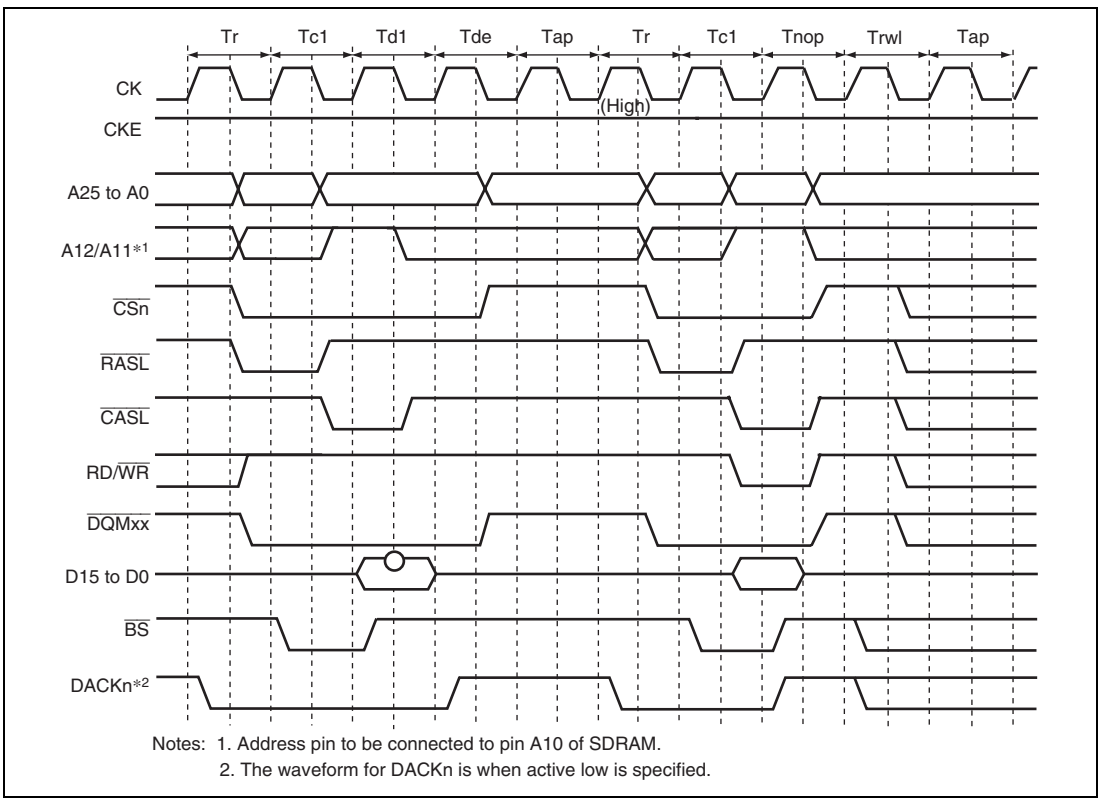


Figure 8.27 Low-Frequency Mode Access Timing

(11) Power-Down Mode

If the PDOWN bit in SDCR is set to 1, the SDRAM is placed in power-down mode by bringing the CKE signal to the low level in the non-access cycle. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note that if an access occurs in power-down mode, a cycle of overhead occurs because a cycle is needed to assert the CKE in order to cancel power-down mode.

Figure 8.28 shows the access timing in power-down mode.

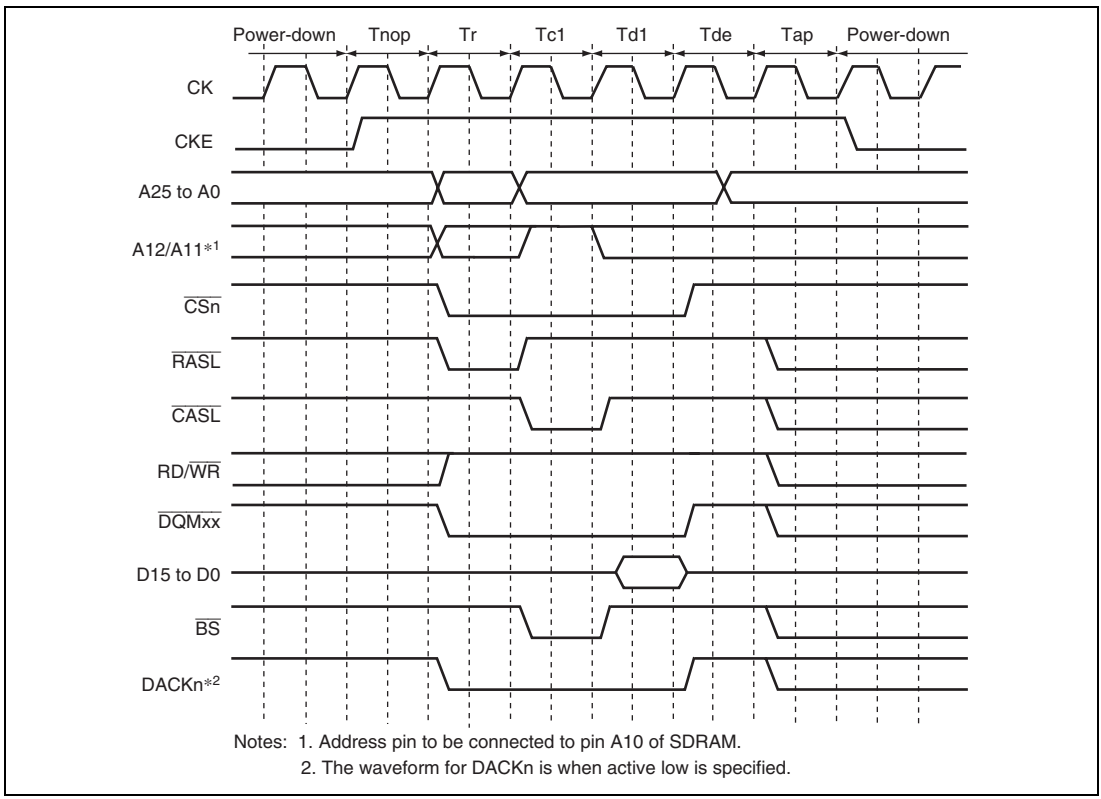


Figure 8.28 Power-Down Mode Access Timing

(12) Power-On Sequence

In order to use SDRAM, mode setting must first be made for SDRAM after waiting for 100 μs or a longer period after powering on. This 100-μs or longer period should be obtained by a power-on reset generating circuit or software.

To perform SDRAM initialization correctly, the bus state controller registers must first be set, followed by a write to the SDRAM mode register. In SDRAM mode register setting, the address signal value at that time is latched by a combination of the $\overline{CSn}$, $\overline{RASL}$, $\overline{CASL}$, and RD/$\overline{WR}$ signals. If the value to be set is X, the bus state controller provides for value X to be written to the SDRAM mode register by performing a write to address H'FFFC4000 + X for area 2 SDRAM, and to address H'FFFC5000 + X for area 3 SDRAM. In this operation the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write, CAS latency 2 to 3, wrap type = sequential, and burst length 1 supported by the LSI, arbitrary data is written in a byte-size access to the addresses shown in table 8.13. In this time 0 is output at the external address pins of A12 or later.

Table 8.13 Access Address in SDRAM Mode Register Write

- Setting for Area 2

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460

Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060

- Setting for Area 3

  Burst read/single write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460

  Burst read/burst write (burst length 1):

Data Bus Width	CAS Latency	Access Address	External Address Pin
16 bits	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060

Mode register setting timing is shown in figure 8.29. A PALL command (all bank pre-charge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register write command) is finally issued. Idle cycles, of which number is specified by the WTRP1 and WTRP0 bits in CS3WCR, are inserted between the PALL and the first REF. Idle cycles, of which number is specified by the WTRC1 and WTRC0 bits in CS3WCR, are inserted between REF and REF, and between the 8th REF and MRS. Idle cycles, of which number is one or more, are inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer to the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer than the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.

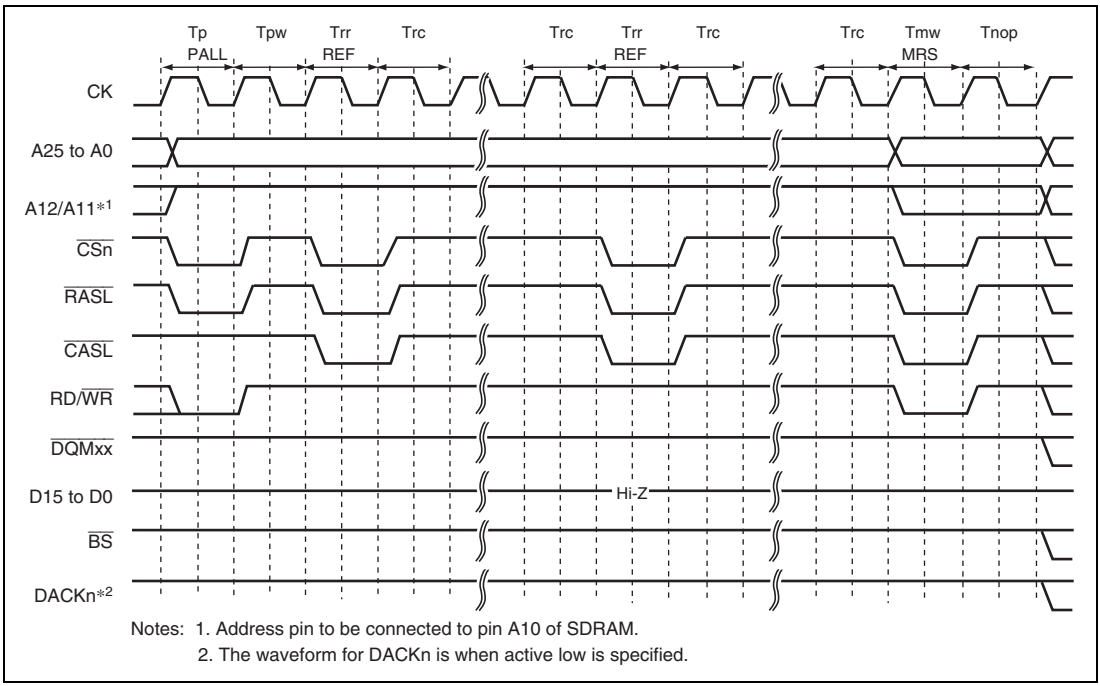


Figure 8.29 SDRAM Mode Write Timing (Based on JEDEC)

(13) Low-Power SDRAM

The low-power SDRAM can be accessed using the same protocol as the normal SDRAM.

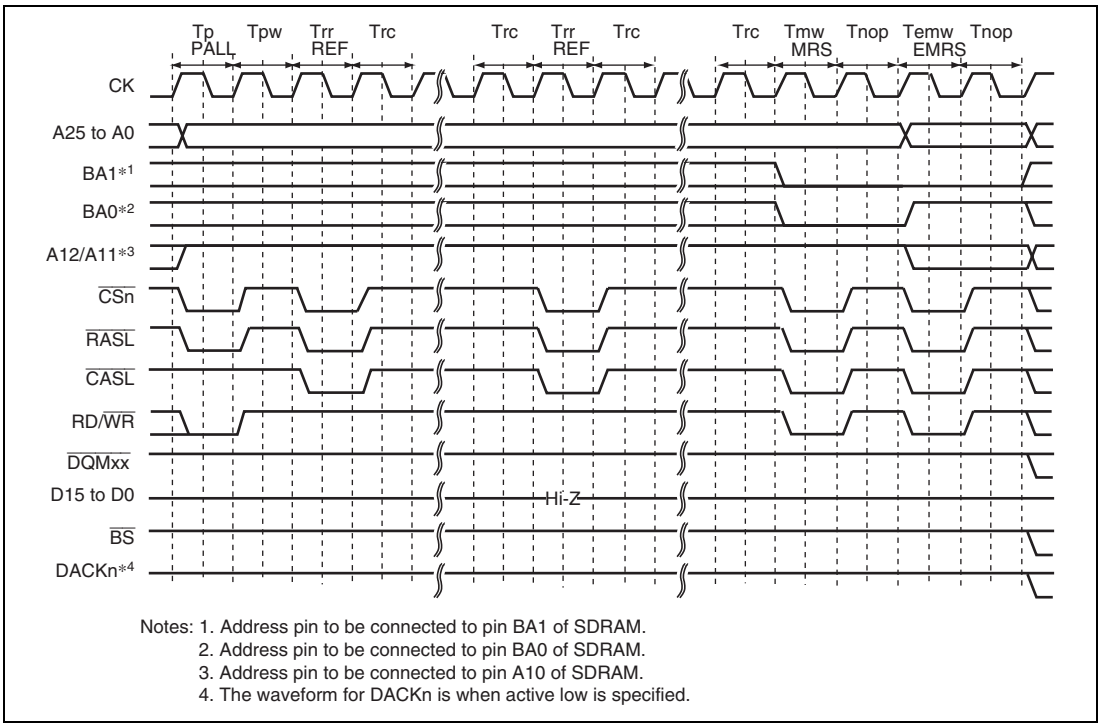
The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which there is data in a work area other than the specific area can be lost without severe repercussions.

The low-power SDRAM supports the extension mode register (EMRS) in addition to the mode registers as the normal SDRAM. This LSI supports issuing of the EMRS command.

The EMRS command is issued according to the conditions specified in table below. For example, if data H'0YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are H'0000XX0 and H'YYYYYYY, respectively. If data H'1YYYYYYY is written to address H'FFFC5XX0 in longword, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 8.14 Output Addresses when EMRS Command Is Issued

Command to be Issued	Access Address	Access Data	Write Access Size	MRS Command Issue Address	EMRS Command Issue Address
CS2 MRS	H'FFFC4XX0	H'********	16 bits	H'0000XX0	—
CS3 MRS	H'FFFC5XX0	H'********	16 bits	H'0000XX0	—
CS2 MRS + EMRS (with refresh)	H'FFFC4XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (with refresh)	H'FFFC5XX0	H'0YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS2 MRS + EMRS (without refresh)	H'FFFC4XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY
CS3 MRS + EMRS (without refresh)	H'FFFC5XX0	H'1YYYYYYY	32 bits	H'0000XX0	H'YYYYYYY



Notes: 1. Address pin to be connected to pin BA1 of SDRAM.
2. Address pin to be connected to pin BA0 of SDRAM.
3. Address pin to be connected to pin A10 of SDRAM.
4. The waveform for DACKn is when active low is specified.

Figure 8.30 EMRS Command Issue Timing

- Deep power-down mode

  The low-power SDRAM supports deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit in the SDCR is set to 1 while the DEEP and RFSH bits in the SDCR are set to 1, the low-power SDRAM enters deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel deep power-down mode. Before executing an access after returning from deep power-down mode, the power-up sequence must be re-executed.

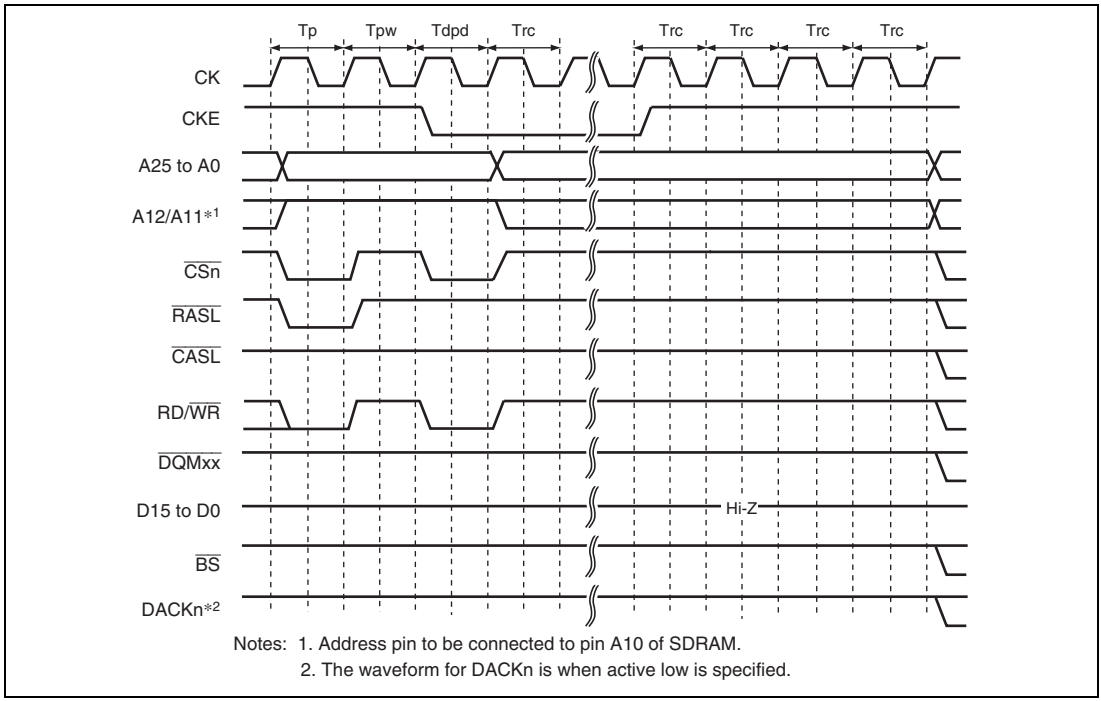


Figure 8.31 Deep Power-Down Mode Transition Timing

8.5.7 Burst ROM (Clock Asynchronous) Interface

The burst ROM (clock asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called burst mode or page mode. In a burst ROM (clock asynchronous) interface, basically the same access as the normal space is performed, but the 2nd and subsequent access cycles are performed only by changing the address, without negating the $\overline{RD}$ signal at the end of the 1st cycle. In the 2nd and subsequent access cycles, addresses are changed at the falling edge of the CK.

For the 1st access cycle, the number of wait cycles specified by the W3 to W0 bits in CSnWCR is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the W1 to W0 bits in CSnWCR is inserted.

In the access to the burst ROM (clock asynchronous), the $\overline{BS}$ signal is asserted only to the first access cycle. An external wait input is valid only to the first access cycle.

In the single access or write access that does not perform the burst operation in the burst ROM (clock asynchronous) interface, access timing is same as a normal space. In addition, there are some restrictions on 16-byte write access. For details, see section 8.6, Usage Notes.

Table 8.15 lists a relationship between bus width, access size, and the number of bursts. Figure 8.32 shows a timing chart.

Table 8.15 Relationship between Bus Width, Access Size, and Number of Bursts

Bus Width	Access Size	CSnWCR. BST[1:0] Bits	Number of Bursts	Access Count
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes	00	16	1
		01	4	4
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes	00	8	1
		01	2	4
		10*	4	2
			2, 4, 2	3

Note: * When the bus width is 16 bits, the access size is 16 bits, and the BST[1:0] bits in CSnWCR are 10, the number of bursts and access count depend on the access start address. At address H'xxx0 or H'xxx8, 4-4 burst access is performed. At address H'xxx4 or H'xxxC, 2-4-2 burst access is performed.

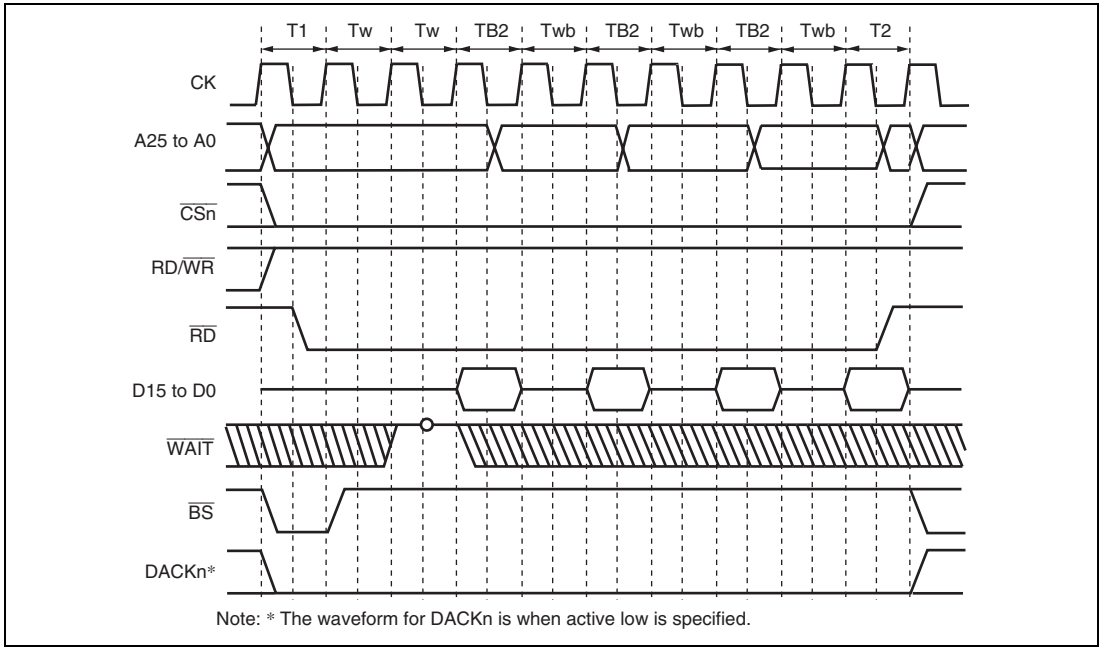


Figure 8.32 Burst ROM Access Timing (Clock Asynchronous)
(Bus Width = 32 Bits, 16-Byte Transfer (Number of Burst 4), Wait Cycles Inserted in First Access = 2, Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

8.5.8 SRAM Interface with Byte Selection

The SRAM interface with byte selection is for access to an SRAM which has a byte-selection pin ($\overline{\text{WEn}}$). This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB and LB.

When the BAS bit in CSnWCR is cleared to 0 (initial value), the write access timing of the SRAM interface with byte selection is the same as that for the normal space interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the $\overline{\text{WEn}}$ pin, which is different from that for the normal space interface. The basic access timing is shown in figure 8.33. In write access, data is written to the memory according to the timing of the byte-selection pin ($\overline{\text{WEn}}$). For details, please refer to the Data Sheet for the corresponding memory.

If the BAS bit in CSnWCR is set to 1, the $\overline{\text{WEn}}$ pin and RD/$\overline{\text{WR}}$ pin timings change. Figure 8.34 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/$\overline{\text{WR}}$). The data hold timing from RD/$\overline{\text{WR}}$ negation to data write must be acquired by setting the HW1 and HW0 bits in CSnWCR. Figure 8.35 shows the access timing when a software wait is specified.

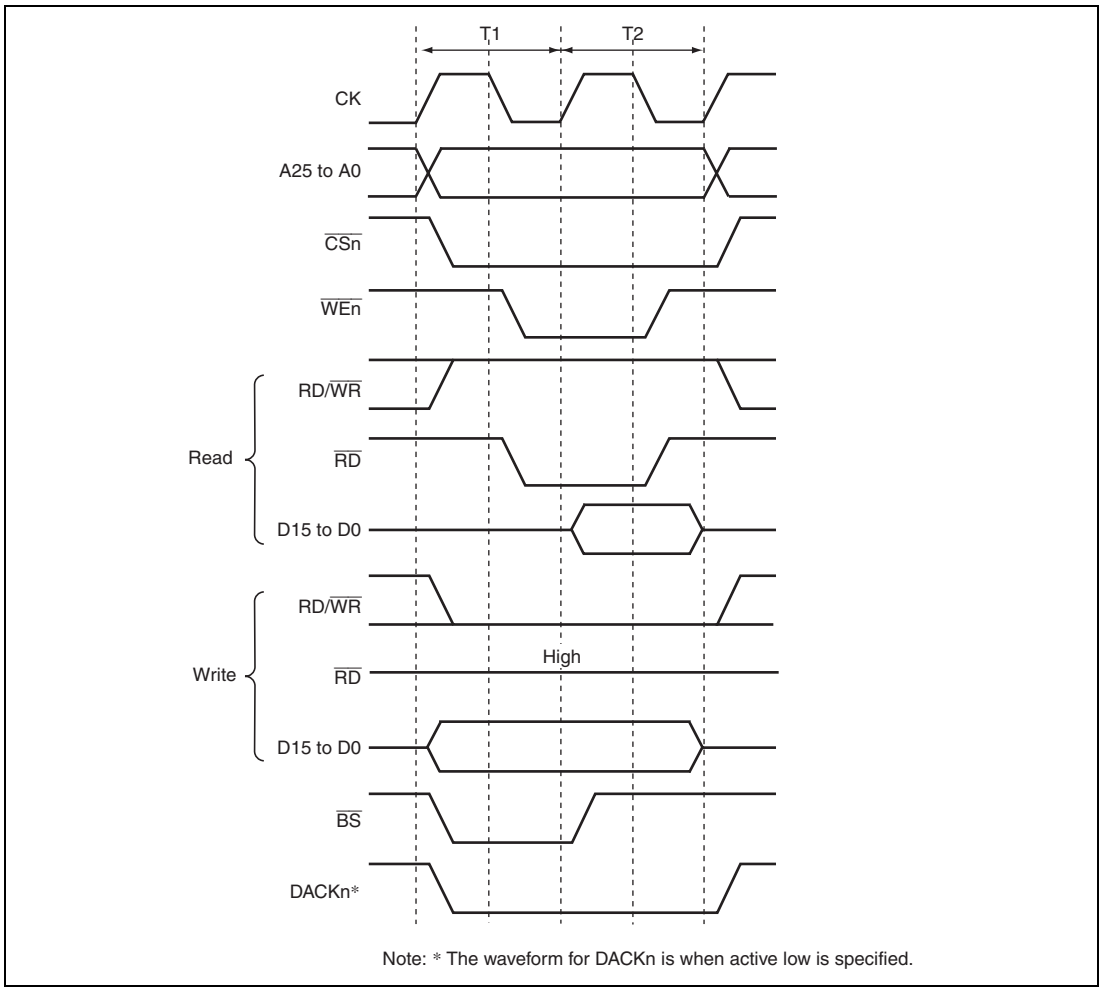


Figure 8.33 Basic Access Timing for SRAM with Byte Selection (BAS = 0)

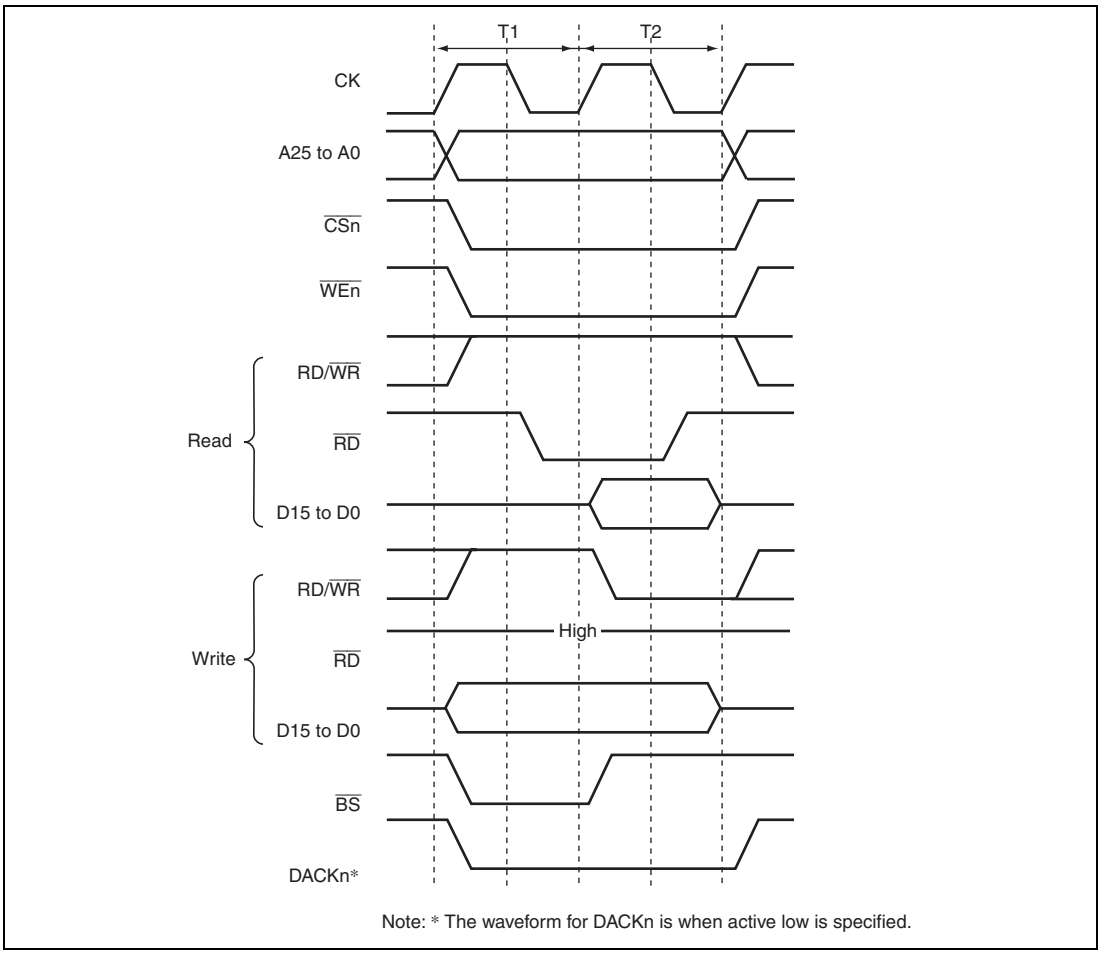


Figure 8.34 Basic Access Timing for SRAM with Byte Selection (BAS = 1)

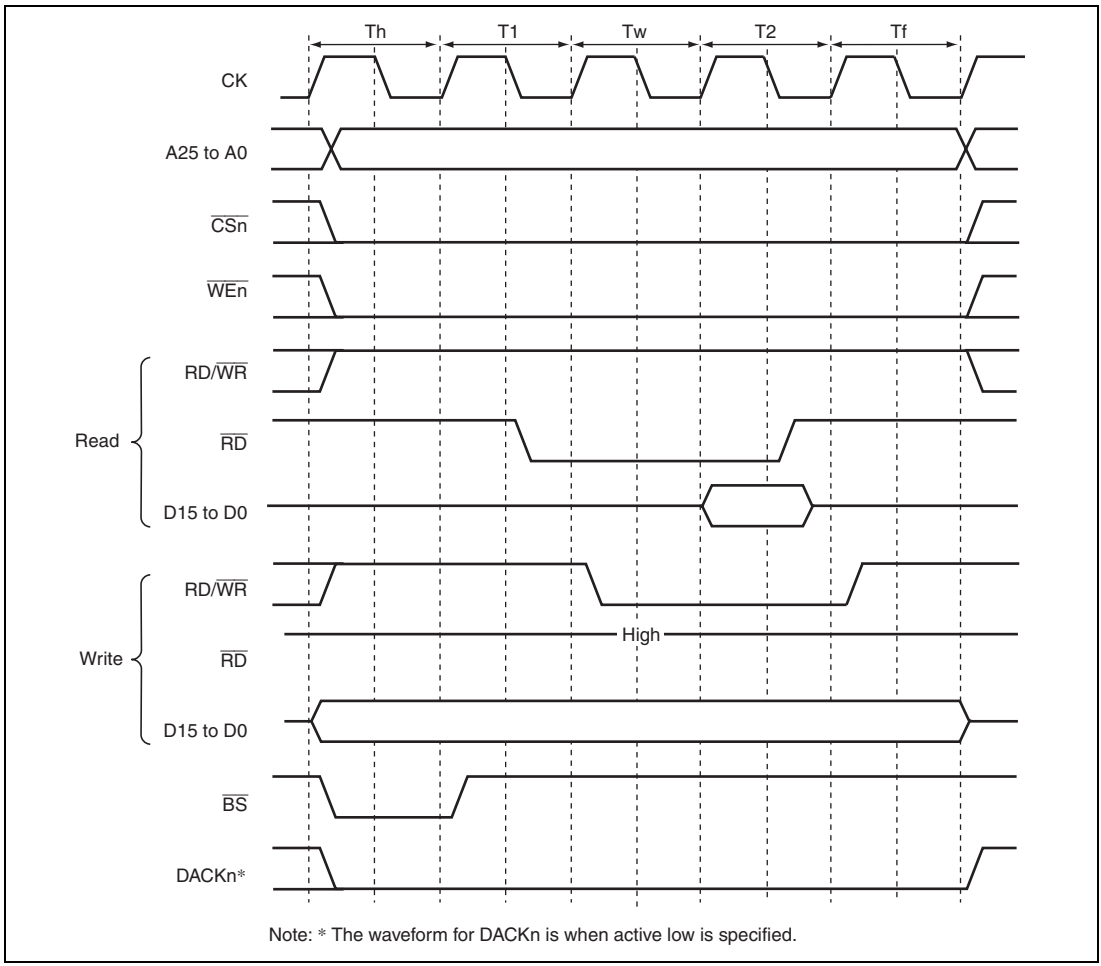


Figure 8.35 Wait Timing for SRAM with Byte Selection (BAS = 1) (SW[1:0] = 01, WR[3:0] = 0001, HW[1:0] = 01)

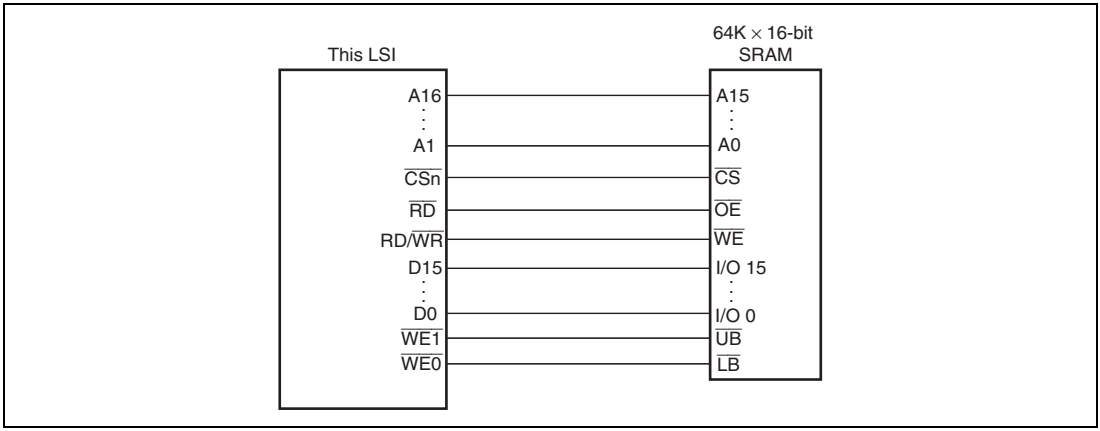


Figure 8.36 Example of Connection with 16-Bit Data-Width SRAM with Byte Selection

8.5.9 Burst ROM (Clock Synchronous) Interface

The burst ROM (clock synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as a normal space. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM (clock synchronous) is accessed, the $\overline{BS}$ signal is asserted only for the first access cycle and an external wait input is also valid for the first access cycle.

If the bus width is 16 bits, the burst length must be specified as 8. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a longword access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, using 16-byte read by the DMA is recommended. The burst ROM interface performs write access in the same way as normal space access.

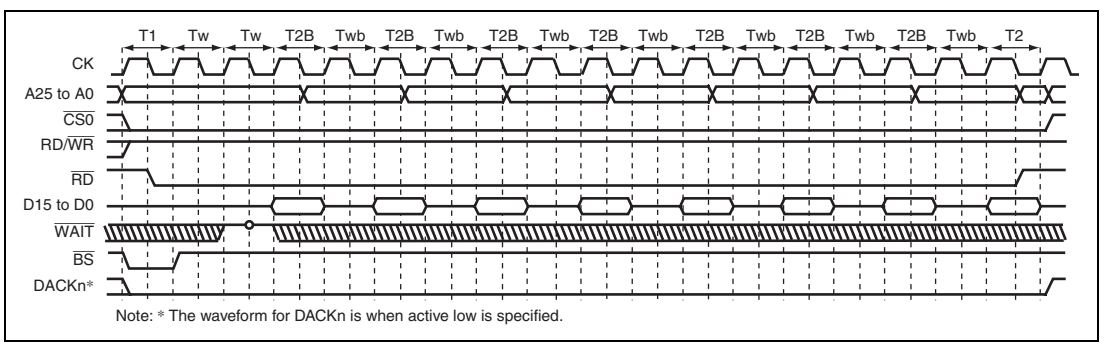


Figure 8.37 Burst ROM Access Timing (Clock Synchronous)
(Burst Length = 8, Wait Cycles Inserted in First Access = 2,
Wait Cycles Inserted in Second and Subsequent Access Cycles = 1)

8.5.10 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the data buffer often collides with the next data access when the read operation from devices with slow access speed is completed. As a result of these collisions, the reliability of the device is low and malfunctions may occur. A function that avoids data collisions by inserting idle (wait) cycles between continuous access cycles has been newly added.

The number of wait cycles between access cycles can be set by the WM bit in CSnWCR, bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR, and bits DMAIW2 to DMAIW0 and DMAIWA in CMNCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space
6. Data output from an external device caused by DMA single address transfer is followed by data output from another device that includes this LSI (DMAIWA = 0)
7. Data output from an external device caused by DMA single address transfer is followed by any type of access (DMAIWA = 1)

For the specification of the number of idle cycles between access cycles described above, refer to the description of each register.

Besides the idle cycles between access cycles specified by the registers, idle cycles must be inserted to interface with the internal bus or to obtain the minimum pulse width for a multiplexed pin ($\overline{\text{WEn}}$). The following gives detailed information about the idle cycles and describes how to estimate the number of idle cycles.

The number of idle cycles on the external bus from $\overline{\text{CSn}}$ negation to $\overline{\text{CSn}}$ or $\overline{\text{CSm}}$ assertion is described below.

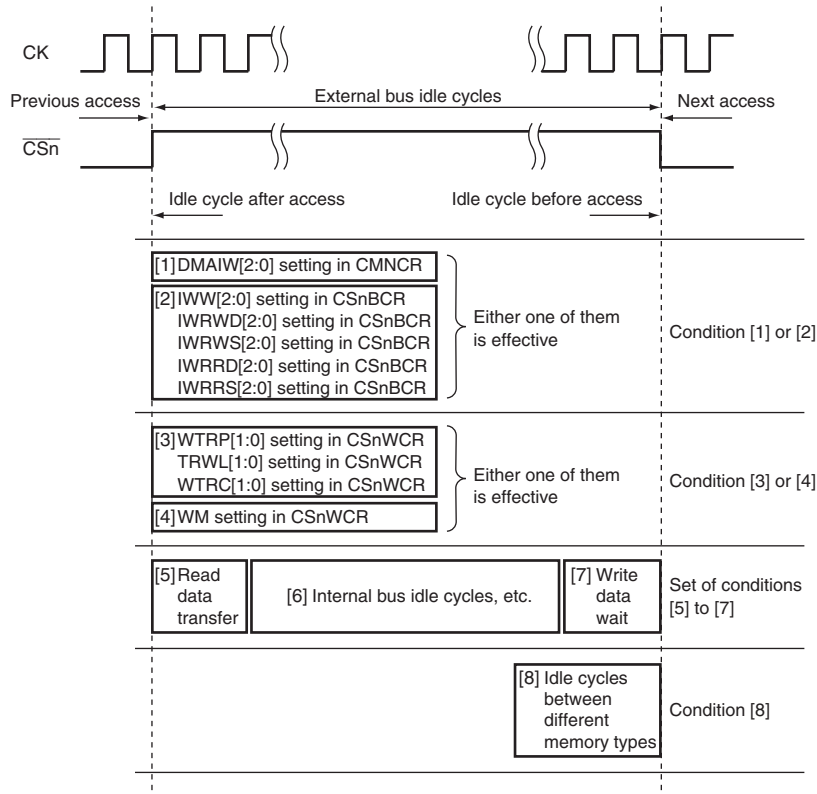
There are eight conditions that determine the number of idle cycles on the external bus as shown in table 8.16. The effects of these conditions are shown in figure 8.38.

Table 8.16 Conditions for Determining Number of Idle Cycles

No.	Condition	Description	Range	Note
(1)	DMAIW[2:0] in CMNCR	These bits specify the number of idle cycles for DMA single address transfer. This condition is effective only for single address transfer and generates idle cycles after the access is completed.	0 to 12	When 0 is specified for the number of idle cycles, the DACK signal may be asserted continuously. This causes a discrepancy between the number of cycles detected by the device with DACK and the DMAC transfer count, resulting in a malfunction.
(2)	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access other than single address transfer. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS1 space followed by reading other CS space, the bits IWRRD[2:0] in CS1BCR should be set to B'100 to specify six or more idle cycles. This condition is effective only for access cycles other than single address transfer and generates idle cycles after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
(3)	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait cycles and idle cycles between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle cycles after the access is completed	0 to 3	Specify these bits in accordance with the specification of the target SDRAM.
(4)	WM in CSnWCR	This bit enables or disables external $\overline{\text{WAIT}}$ pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external $\overline{\text{WAIT}}$ enabled), one idle cycle is inserted to check the external $\overline{\text{WAIT}}$ pin input after the access is completed. When this bit is set to 1 (disabled), no idle cycle is generated.	0 or 1	

No.	Condition	Description	Range	Note
(5)	Read data transfer cycle	One idle cycle is inserted after a read access is completed. This idle cycle is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HM[1:0] bits in CSnWCR are not B'00.	0 or 1	One idle cycle is always generated after a read cycle with SDRAM interface.
(6)	Internal bus idle cycles, etc.	External bus access requests from the CPU or DMAC and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the BSC when the access size is larger than the external data bus width.	0 or larger	The number of internal bus idle cycles may not become 0 depending on the Iϕ:Bϕ clock ratio. Tables 8.17 and 8.18 show the relationship between the clock ratio and the minimum number of internal bus idle cycles.
(7)	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles are frequently available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through CSnBCR.
(8)	Idle cycles between different memory types	To ensure the minimum pulse width on the signal-multiplexed pins, idle cycles may be inserted before access after memory types are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2.5	The number of idle cycles depends on the target memory types. See table 8.19.

In the above conditions, a total of four conditions, that is, condition (1) or (2) (either one is effective), condition (3) or (4) (either one is effective), a set of conditions (5) to (7) (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition (8) are generated at the same time. The maximum number of idle cycles among these four conditions become the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition (1) or (2).



Note: A total of four conditions (condition [1] or [2], condition [3] or [4], a set of conditions [5] to [7], and condition [8]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 8.38 Idle Cycle Conditions

Table 8.17 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 8.18 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 8.19 Number of Idle Cycles Inserted between Access Cycles to Different Memory Types

	Next Cycle							
Previous Cycle	**SRAM**	**Burst ROM (Asynchronous)**	**MPX-I/O**	**Byte SRAM (BAS = 0)**	**Byte SRAM (BAS = 1)**	**SDRAM**	**SDRAM (Low-Frequency Mode)**	**Burst ROM (Synchronous)**
SRAM	0	0	1	0	1	1	1.5	0
Burst ROM (asynchronous)	0	0	1	0	1	1	1.5	0
MPX-I/O	1	1	0	1	1	1	1.5	1
Byte SRAM (BAS = 0)	0	0	1	0	1	1	1.5	0
Byte SRAM (BAS = 1)	1	1	2	1	0	0	1.5	1
SDRAM	1	1	2	1	0	0	—	1
SDRAM (low-frequency mode)	1.5	1.5	2.5	1.5	0.5	—	1	1.5
Burst ROM (synchronous)	0	0	1	0	1	1	1.5	0

Figure 8.39 shows sample estimation of idle cycles between access cycles. In the actual operation, the idle cycles may become shorter than the estimated value due to the write buffer effect or may become longer due to internal bus idle cycles caused by stalling in the pipeline due to CPU instruction execution or CPU register conflicts. Please consider these errors when estimating the idle cycles.

Sample Estimation of Idle Cycles between Access Cycles

This example estimates the idle cycles for data transfer from the CS1 space to CS2 space by CPU access. Transfer is repeated in the following order: CS1 read → CS1 read → CS2 write → CS2 write → CS1 read → ...

- Conditions
  The bits for setting the idle cycles between access cycles in CS1BCR and CS2BCR are all set to 0.
  In CS1WCR and CS2WCR, the WM bit is set to 1 (external $\overline{\text{WAIT}}$ pin disabled) and the HW[1:0] bits are set to 00 ($\overline{\text{CS}}$ negation is not extended).
  Iϕ:Bϕ is set to 4:1, and no other processing is done during transfer.
  For both the CS1 and CS2 spaces, normal SRAM devices are connected, the bus width is 32 bits, and access size is also 32 bits.

The idle cycles generated under each condition are estimated for each pair of access cycles. In the following table, R indicates a read cycle and W indicates a write cycle.

Condition	R → R	R → W	W → W	W → R	Note
[1] or [2]	0	0	0	0	CSnBCR is set to 0.
[3] or [4]	0	0	0	0	The WM bit is set to 1.
[5]	1	1	0	0	Generated after a read cycle.
[6]	0	2	2	0	See the Iϕ:Bϕ = 4:1 columns in table 8.17.
[7]	0	1	0	0	No idle cycle is generated for the second time due to the write buffer effect.
[5] + [6] + [7]	0	4	2	0	
[8]	0	0	0	0	Value for SRAM → SRAM access
Estimated idle cycles	1	4	2	0	Maximum value among conditions [1] or [2], [3] or [4], [5] + [6] + [7], and [8]
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 [6] is estimated as 0 but actually an internal idle cycle is generated due to execution of a loop condition check instruction.

Figure 8.39 Comparison between Estimated Idle Cycles and Actual Value

8.5.11 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.

With the SDRAM interface, all bank pre-charge commands (PALLs) are issued when active banks exist and the bus is released after completion of a PALL command.

The bus sequence is as follows. The address bus and data bus are placed in a high-impedance state synchronized with the rising edge of CK. The bus mastership enable signal is asserted 0.5 cycles after the above timing, synchronized with the falling edge of CK. The bus control signals ($\overline{\text{BS}}$, $\overline{\text{CSn}}$, $\overline{\text{RASL}}$, $\overline{\text{CASL}}$, CKE, $\overline{\text{DQMxx}}$, $\overline{\text{WEn}}$, $\overline{\text{RD}}$, and RD/$\overline{\text{WR}}$) are placed in the high-impedance state at subsequent rising edges of CK. Bus request signals are sampled at the falling edge of CKIO. Note that CKE, $\overline{\text{RASL}}$, and $\overline{\text{CASL}}$ can continue to be driven at the previous value even in the bus-released state by setting the HIZCNT bit in CMNCR.

The sequence for reclaiming the bus mastership from an external device is described below. 1.5 cycles after the negation of $\overline{\text{BREQ}}$ 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 8.40 shows the bus arbitration timing.

When it is necessary to refresh SDRAM while releasing the bus mastership, the bus mastership should be returned using the $\overline{\text{REFOUT}}$ signal. For details on the selection of $\overline{\text{REFOUT}}$, see section 19, Pin Function Controller (PFC). The $\overline{\text{REFOUT}}$ signal is kept asserting at low level until the bus mastership is acquired. The $\overline{\text{BREQ}}$ signal is negated by asserting the $\overline{\text{REFOUT}}$ signal and the bus mastership is returned from the external device. If the bus mastership is not returned for a refreshing period or longer, the contents of SDRAM cannot be guaranteed because a refreshing cannot be executed.

While releasing the bus mastership, the SLEEP instruction (to enter sleep mode or standby mode), as well as a manual reset, cannot be executed until the LSI obtains the bus mastership.

The $\overline{\text{BREQ}}$ 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.

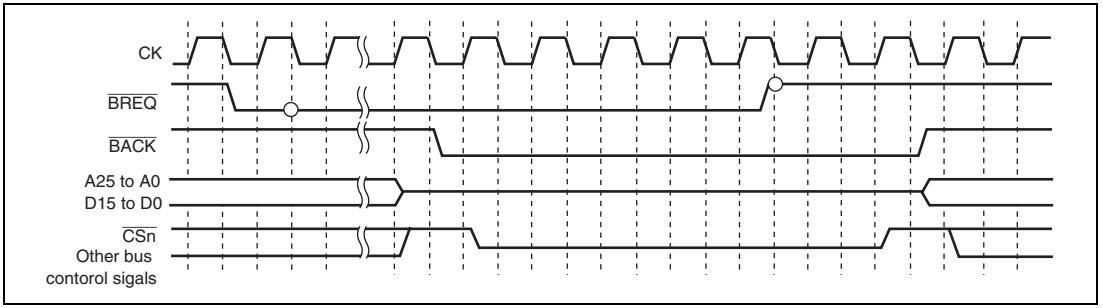


Figure 8.40 Bus Arbitration Timing (Clock Mode 7 or CMNCR.HIZCNT = 1)

8.5.12 Others

(1) Reset

The bus state controller (BSC) can be initialized completely only at power-on reset. At power-on reset, all signals are negated and data output buffers are turned off regardless of the bus cycle state after the internal reset is synchronized with the internal clock. All control registers are initialized. In standby, sleep, and manual reset, control registers of the bus state controller are not initialized. At manual reset, only the current bus cycle being executed is completed. Since the RTCNT continues counting up during manual reset signal assertion, a refresh request occurs to initiate the refresh cycle.

(2) Access from the Side of the LSI Internal Bus Master

Since the bus state controller (BSC) incorporates a one-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

Access to the on-chip peripheral module registers from the internal bus requires 2 or more cycles of the peripheral module clock (Pϕ). When the CPU writes to an on-chip peripheral register, however, the CPU can execute the following instructions without waiting for the register write to complete.

This section describes the case where the system switches to software standby mode to reduce power consumption as an example. In this case, the code sets the STBCR register STBY bit to 1

and then executes a SLEEP instruction. The code must, however, perform a dummy read of the STBCR register before executing the SLEEP instruction. If the dummy read is not performed, the CPU will execute the SLEEP instruction before the STBY bit is set to 1, and the system will not switch to the intended software standby mode, but rather will switch to sleep mode. The dummy read of the STBCR register is required to wait for the write to the STBY bit to complete.

In other cases as well, application code should perform a dummy read of the same register after a register write instruction and only then execute the following instructions for the intended purpose to assure that the changes due to internal register writes are reflected when the following instructions are executed as in this example.

The table below lists the number of access cycles required for CPU accesses to the on-chip peripheral module registers.

Table 8.20 On-Chip Peripheral Module Register Access Cycle Counts

	Access Cycles
Write	$(2+n) \times I\phi + (1+m) \times B\phi + 2 \times P\phi$
Read	$(2+n) \times I\phi + (1+m) \times B\phi + 2 \times P\phi + (2+I) \times I\phi$

Note: These are the numbers of cycles when the instruction is executed from internal ROM or internal RAM.
When Iϕ:Bϕ is 1:1; n = 0, I = 0
When Iϕ:Bϕ is 2:1; n = 0 or 1, I = 1
When Iϕ:Bϕ is 4:1; n = 0 to 3, I = 2
When Iϕ:Bϕ is 8:1; n = 0 to 7, I = 2
When Bϕ:Pϕ is 1:1; m = 0
When Bϕ:Pϕ is 2:1; m = 0 or 1
When Bϕ:Pϕ is 4:1; m = 0 to 3
Note that n and m depend on the internal execution state.

This product adopts synchronized logic and has a hierarchical bus structure. Data input and output for each of the busses is synchronized with the rising edge of the Iϕ clock for the C bus, the Bϕ clock for the I bus, and the Pϕ clock for the peripheral bus.

Figure 8.41 shows an example of the write timing to the peripheral bus when the relationship between the clocks is Iϕ:Bϕ:Pϕ = 4:4:1. Data is output in synchronization with Iϕ to the C bus, to which the CPU is connected. When Iϕ:Bϕ is 1:1, 2 × Iϕ + Bϕ periods are required for data transfers from the C bus to the I bus. For transfers from the I bus to the peripheral bus when Bϕ:Pϕ is 4:1, since there are four clock cycles during a single Pϕ clock period, the timing with which the data is placed on the peripheral bus is as follows: there are four timings for Pϕ × 1, and up to 4 Bϕ periods are required for the Pϕ rising edge, which is the timing for transfers from the I bus to the

peripheral bus (the example in figure 8.41 is for 4 × Bϕ). Therefore, when Bϕ:Pϕ is 4:1, data is transferred from the I bus to the peripheral bus in time (1+m) × Bϕ, where m = 0 to 3 periods. Note that the relationship between the timing with which the data appears on the I bus and the Pϕ rising edge depends on the program execution state. In figure 8.41, since n = 0 and m = 3, the access time will be 2 × Iϕ + 4 × Bϕ + 2 × Pϕ.

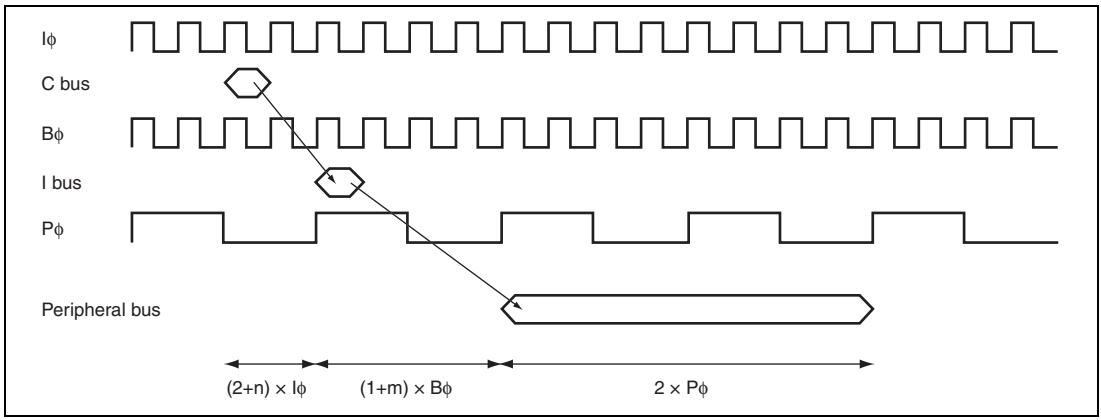


Figure 8.41 Internal Peripheral I/O Register Timing when Iϕ:Bϕ:Pϕ = 4:4:1

Figure 8.42 shows an example of the write timing to the peripheral bus when the relationship between the clocks is Iϕ:Bϕ:Pϕ = 4:2:1. Although transfers from the C bus to the peripheral bus are performed the same way for write, for read, the value read from the peripheral bus must be transferred to the CPU. Although the transfers from the peripheral bus to the I bus and from the I bus to the C bus are all performed on the corresponding bus clock rising edge, since Iϕ ≥ Bϕ ≥ Pϕ, (2 + 1) × Iϕ periods are actually required. In the example in figure 8.42, since n = 1, m = 1, and i = 1, the access period will be 3 × Iϕ + 2 × Bϕ + 2 × Pϕ + 3 × Iϕ.

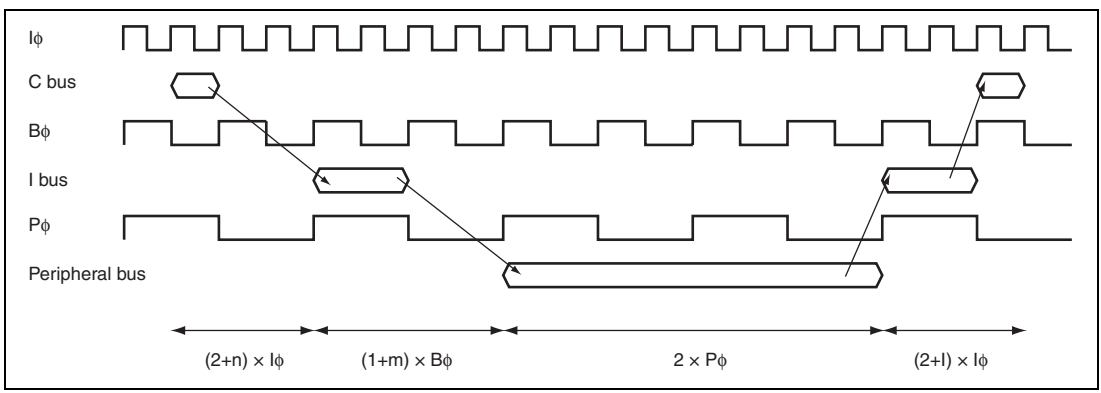


Figure 8.42 Internal Peripheral I/O Register Timing when Iϕ:Bϕ:Pϕ = 4:2:1

8.6 Usage Notes

8.6.1 Burst ROM Interface

When the burst ROM interface (clock asynchronous) is used and the following three conditions are met, read/write access from the external bus space immediately after write access may be invalid.

1. The 16-bit bus width is used for the burst ROM interface (clock asynchronous). (The CSnBCR.TYPE[2:0] setting is B'001 and the CSnWCR.BSZ[1:0] setting is B'10)
2. The burst length is specified as 4. (The CSnWCR.BST[1:0] setting is B'10)
3. Write-back is performed with operand cache or 16-byte write access is performed with the DMAC for the burst ROM interface set as above.

Section 9 Direct Memory Access Controller (DMAC)

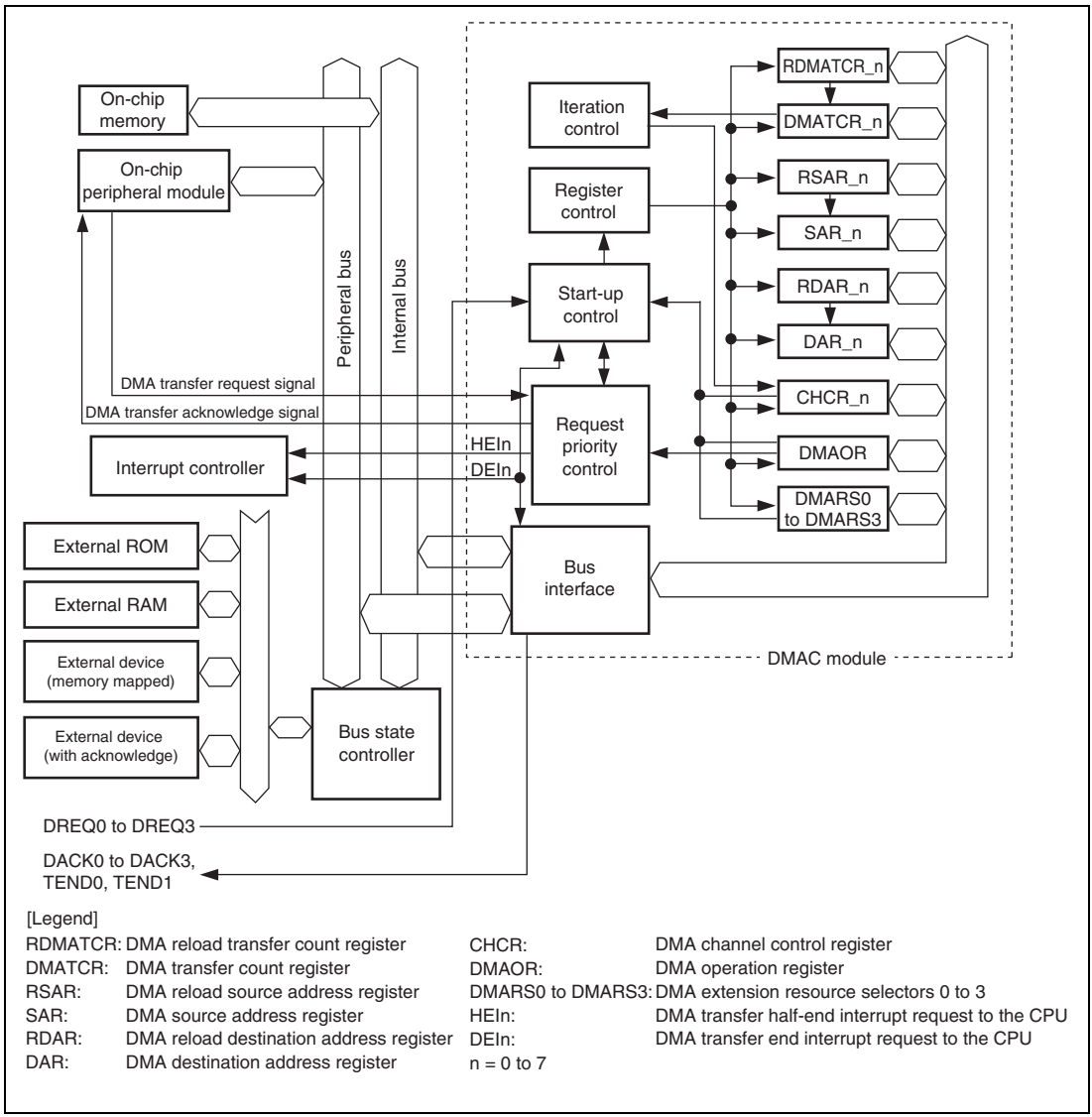
The DMAC can be used in place of the CPU to perform high-speed transfers between external devices that have DACK (transfer request acknowledge signal), external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

9.1 Features

- Number of channels: Eight channels (channels 0 to 7) selectable
  Four channels (channels 0 to 3) can 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.
  — Eight SCIF sources, two IIC3 sources, one A/D converter source, five MTU2 sources, and two CMT sources
- 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 9.1 shows the block diagram of the DMAC.



[Legend]

RDMATCR:	DMA reload transfer count register	CHCR:	DMA channel control register
DMATCR:	DMA transfer count register	DMAOR:	DMA operation register
RSAR:	DMA reload source address register	DMARS0 to DMARS3:	DMA extension resource selectors 0 to 3
SAR:	DMA source address register	HEIn:	DMA transfer half-end interrupt request to the CPU
RDAR:	DMA reload destination address register	DEIn:	DMA transfer end interrupt request to the CPU
DAR:	DMA destination address register	n = 0 to 7	

Figure 9.1 Block Diagram of DMAC

9.2 Input/Output Pins

The external pins for DMAC are described below. Table 9.1 lists the configuration of the pins that are connected to external bus. DMAC has pins for four channels (channels 0 to 3) for external bus use.

Table 9.1 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

9.3 Register Descriptions

The DMAC has the registers listed in table 9.2. 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 9.2 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

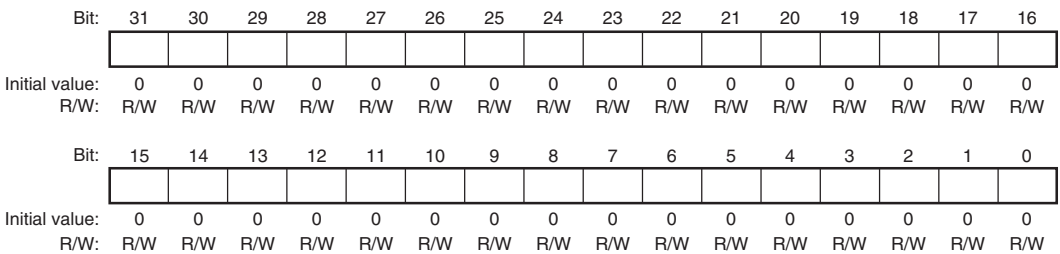
Notes: 1. For the HE and TE bits in CHCRn, only 0 can be written to clear the flags after 1 is read.
2. For the AE and NMIF bits in DMAOR, only 0 can be written to clear the flags after 1 is read.

9.3.1 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.



9.3.2 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

9.3.3 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

9.3.4 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 SCIF or IIC3 is selected for the transfer request source, this bit (TC) must not be set to 1. 0: Transmits data once by one transfer request 1: Transmits data for the count specified in DMATCR by one transfer request
30, 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
28	RLD	0	R/W	Reload Function Enable or Disable Enables or disables the reload function. 0: Disables the reload function 1: Enables the reload function
27 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23	DO	0	R/W	DMA Overrun Selects whether DREQ is detected by overrun 0 or by overrun 1. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 and CHCR_7; it is always read as 0 and the write value should always be 0. 0: Detects DREQ by overrun 0 1: Detects DREQ by overrun 1
22	TL	0	R/W	Transfer End Level Specifies the TEND signal output is high active or low active. This bit is valid only in CHCR_0 and CHCR_1. This bit is reserved in CHCR_2 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: Low-active output from TEND 1: High-active output from TEND
21, 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Descriptions
19	HE	0	R/(W)*	Half-End Flag This bit is set to 1 when the transfer count reaches half of the DMATCR value that was specified before transfer starts. If DMA transfer ends because of an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR before the transfer count reaches half of the initial DMATCR value, the HE bit is not set to 1. If DMA transfer ends due to an NMI interrupt, a DMA address error, or clearing of the DE bit or the DME bit in DMAOR after the HE bit is set to 1, the bit remains set to 1. To clear the HE bit, write 0 to it after HE = 1 is read. 0: DMATCR > (DMATCR set before transfer starts)/2 during DMA transfer or after DMA transfer is terminated [Clearing condition] • Writing 0 after reading HE = 1. 1: DMATCR ≤ (DMATCR set before transfer starts)/2
18	HIE	0	R/W	Half-End Interrupt Enable Specifies whether to issue an interrupt request to the CPU when the transfer count reaches half of the DMATCR value that was specified before transfer starts. When the HIE bit is set to 1, the DMAC requests an interrupt to the CPU when the HE bit becomes 1. 0: Disables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2 1: Enables an interrupt to be issued when DMATCR = (DMATCR set before transfer starts)/2

Bit	Bit Name	Initial Value	R/W	Descriptions
17	AM	0	R/W	Acknowledge Mode Specifies whether DACK is output in data read cycle or in data write cycle in dual address mode. In single address mode, DACK is always output regardless of the specification by this bit. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: DACK output in read cycle (dual address mode) 1: DACK output in write cycle (dual address mode)
16	AL	0	R/W	Acknowledge Level Specifies the DACK (acknowledge) signal output is high active or low active. This bit is valid only in CHCR_0 to CHCR_3. This bit is reserved in CHCR_4 to CHCR_7; it is always read as 0 and the write value should always be 0. 0: Low-active output from DACK 1: High-active output from DACK

Bit	Bit Name	Initial Value	R/W	Descriptions
15,14	DM[1:0]	00	R/W	Destination Address Mode These bits select whether the DMA destination address is incremented, decremented, or left fixed. (In single address mode, DM1 and DM0 bits are ignored when data is transferred to an external device with DACK.) 00: Fixed destination address (Setting prohibited in 16-byte transfer) 01: Destination address is incremented (+1 in 8-bit transfer, +2 in 16-bit transfer, +4 in 32-bit transfer, +16 in 16-byte transfer) 10: Destination address is decremented (–1 in 8-bit transfer, –2 in 16-bit transfer, –4 in 32-bit transfer, setting prohibited in 16-byte transfer) 11: Setting prohibited
13, 12	SM[1:0]	00	R/W	Source Address Mode These bits select whether the DMA source address is incremented, decremented, or left fixed. (In single address mode, SM1 and SM0 bits are ignored when data is transferred from an external device with DACK.) 00: Fixed source address (Setting prohibited in 16-byte-unit transfer) 01: Source address is incremented (+1 in byte-unit transfer, +2 in word-unit transfer, +4 in longword-unit transfer, +16 in 16-byte-unit transfer) 10: Source address is decremented (–1 in byte-unit transfer, –2 in word-unit transfer, –4 in longword-unit transfer, setting prohibited in 16-byte-unit transfer) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Descriptions
11 to 8	RS[3:0]	0000	R/W	Resource Select These bits specify which transfer requests will be sent to the DMAC. The changing of transfer request source should be done in the state when DMA enable bit (DE) is set to 0. 0000: External request, dual address mode 0001: Setting prohibited 0010: External request/single address mode External address space → External device with DACK 0011: External request/single address mode External device with DACK → External address space 0100: Auto request 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: DMA extension resource selector 1001: Setting prohibited 1010: Setting prohibited 1011: Setting prohibited 1100: Setting prohibited 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited Note: External request specification is valid only in CHCR_0 to CHCR_3. If a request source is selected in channels CHCR_4 to CHCR_7, no operation will be performed.

Bit	Bit Name	Initial Value	R/W	Descriptions
7	DL	0	R/W	DREQ Level
6	DS	0	R/W	DREQ Edge Select These bits specify the sampling method of the DREQ pin input and the sampling level. These bits are valid only in CHCR_0 to CHCR_3. These bits are reserved in CHCR_4 to CHCR_7; they are always read as 0 and the write value should always be 0. If the transfer request source is specified as an on-chip peripheral module or if an auto-request is specified, the specification by these bits is ignored. 00: DREQ detected in low level 01: DREQ detected at falling edge 10: DREQ detected in high level 11: DREQ detected at rising edge
5	TB	0	R/W	Transfer Bus Mode Specifies bus mode when DMA transfers data. Note that burst mode must not be selected when TC = 0. 0: Cycle steal mode 1: Burst mode
4, 3	TS[1:0]	00	R/W	Transfer Size These bits specify the size of data to be transferred. Select the size of data to be transferred when the source or destination is an on-chip peripheral module register of which transfer size is specified. 00: Byte unit 01: Word unit (two bytes) 10: Longword unit (four bytes) 11: 16-byte unit (four longwords)
2	IE	0	R/W	Interrupt Enable Specifies whether or not an interrupt request is generated to the CPU at the end of the DMA transfer. Setting this bit to 1 generates an interrupt request (DEI) to the CPU when TE bit is set to 1. 0: Disables an interrupt request 1: Enables an interrupt request

Bit	Bit Name	Initial Value	R/W	Descriptions
1	TE	0	R/(W)*	Transfer End Flag This bit is set to 1 when DMATCR becomes 0 and DMA transfer ends. The TE bit is not set to 1 in the following cases. • DMA transfer ends due to an NMI interrupt or DMA address error before DMATCR becomes 0. • DMA transfer is ended by clearing the DE bit and DME bit in DMA operation register (DMAOR). To clear the TE bit, write 0 after reading TE = 1. Even if the DE bit is set to 1 while this bit is set to 1, transfer is not enabled. 0: During the DMA transfer or DMA transfer has been terminated [Clearing condition] • Writing 0 after reading TE = 1 1: DMA transfer ends by the specified count (DMATCR = 0)
0	DE	0	R/W	DMA Enable Enables or disables the DMA transfer. In auto-request mode, DMA transfer starts by setting the DE bit and DME bit in DMAOR to 1. In this case, all of the bits TE, NMIF in DMAOR, and AE must be 0. In an external request or peripheral module request, DMA transfer starts if DMA transfer request is generated by the devices or peripheral modules after setting the bits DE and DME to 1. In this case, however, all of the bits TE, NMIF, and AE must be 0 as in the case of auto-request mode. Clearing the DE bit to 0 can terminate the DMA transfer. 0: DMA transfer disabled 1: DMA transfer enabled

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

9.3.5 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

9.3.6 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

9.3.7 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

9.3.8 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 reading AE = 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 reading NMIF = 1

Bit	Bit Name	Initial Value	R/W	Description
0	DME	0	R/W	DMA Master Enable Enables or disables DMA transfer on all channels. If the DME bit and DE bit in CHCR are set to 1, DMA transfer is enabled. However, transfer is enabled only when the TE bit in CHCR of the transfer corresponding channel, the NMIF bit in DMAOR, and the AE bit are all cleared to 0. Clearing the DME bit to 0 can terminate the DMA transfer on all channels. 0: DMA transfer is disabled on all channels 1: DMA transfer is enabled on all channels

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

If the priority mode bits are modified after a DMA transfer, the channel priority is initialized. If fixed mode 2 is specified, the channel priority is specified as CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7. If fixed mode 1 is specified, the channel priority is specified as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7. If round-robin mode is specified, the transfer end channel is reset.

Table 9.3 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 9.3 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

9.3.9 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 9.4 shows the specifiable combinations.

DMARS can specify transfer requests from eight SCIF sources, two IIC3 sources, one A/D converter source, five MTU2 sources, and two CMT 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 9.4.

Table 9.4 DMARS Settings

Peripheral Module	Setting Value for One Channel ({MID, RID})	MID	RID	Function
SCIF_0 transmitter	H'81	B'100000	B'01	Transmit
SCIF_0 receiver	H'82		B'10	Receive
SCIF_1 transmitter	H'85	B'100001	B'01	Transmit
SCIF_1 receiver	H'86		B'10	Receive
SCIF_2 transmitter	H'89	B'100010	B'01	Transmit
SCIF_2 receiver	H'8A		B'10	Receive
SCIF_3 transmitter	H'8D	B'100011	B'01	Transmit
SCIF_3 receiver	H'8E		B'10	Receive
IIC3 transmitter	H'A1	B'101000	B'01	Transmit
IIC3 receiver	H'A2		B'10	Receive
A/D converter	H'B3	B'101100	B'11	—
MTU2_0	H'E3	B'111000	B'11	—
MTU2_1	H'E7	B'111001	B'11	—
MTU2_2	H'EB	B'111010	B'11	—
MTU2_3	H'EF	B'111011	B'11	—
MTU2_4	H'F3	B'111100	B'11	—
CMT_0	H'FB	B'111110	B'11	—
CMT_1	H'FF	B'111111	B'11	—

When MID or RID other than the values listed in table 9.4 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.

9.4 Operation

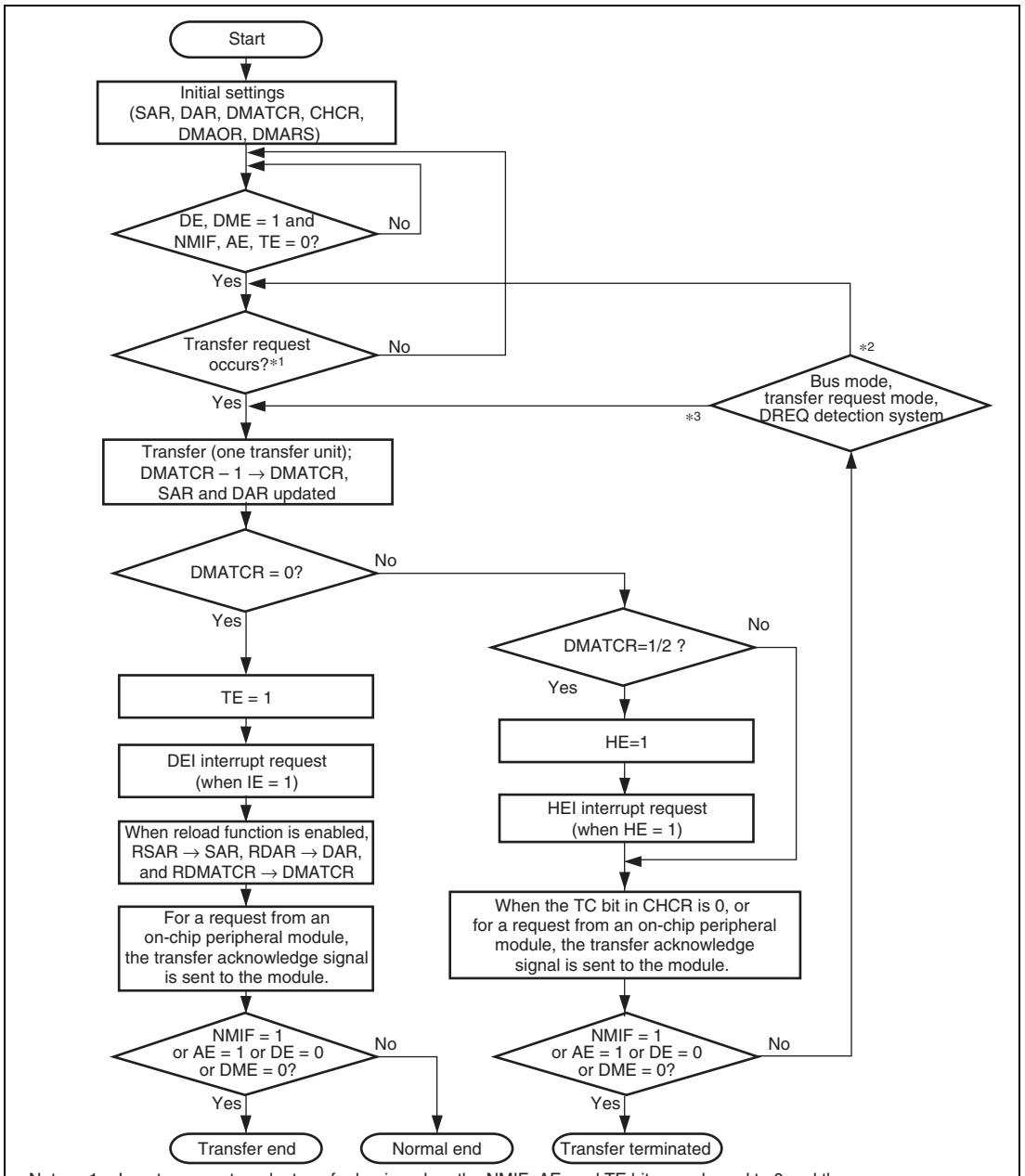
When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in three modes: auto request, external request, and on-chip peripheral module request.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

9.4.1 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 9.2 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 9.2 DMA Transfer Flowchart

9.4.2 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 9.5 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 9.5 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 9.6. The source of the transfer request does not have to be the data transfer source or destination.

Table 9.6 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 9.7 Selecting External Request Detection with DO Bit

CHCR	
DO bit	**External Request**
0	Overrun 0
1	Overrun 1

(3) On-Chip Peripheral Module Request

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

DMA transfer request signals from on-chip peripheral modules to the DMAC include transmit data empty and receive data full requests from the SCIF, A/D conversion end request from the A/D converter, compare match request from the CMT, and data transfer requests from the IIC3 and MTU2.

When a transfer request signal is sent in on-chip peripheral module request mode while DMA transfer is enabled (DE = 1, DME = 1, TE = 0, AE = 0, and NMIF = 0), DMA transfer is performed.

When the transmit data empty from the SCIF is selected, specify the transfer destination as the corresponding SCIF transmit data register. Likewise, when the receive data full from the SCIF is selected, specify the transfer source as the corresponding SCIF receive data register. When a transfer request is made by the A/D converter, the transfer source must be the A/D data register (ADDR). When the IIC3 transmit is selected as the transfer request, the transfer destination must be ICDRT; when the IIC3 reception is selected as the transfer request, the transfer source must be ICDRR. Any address can be specified for data transfer source and destination when a transfer request is sent from the CMT or MTU2.

Table 9.8 Selecting On-Chip Peripheral Module Request Modes with RS3 to RS0 Bits

CHCR	DMARS		DMA Transfer Request Source	DMA Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
RS[3:0]	MID	RID					
1000	100000	01	SCIF_0 transmit	TXI0 (transmit FIFO data empty)	Any	SCFTDR_0	Cycle steal
		10	SCIF_0 receive	RXI0 (receive FIFO data full)	SCFRDR_0	Any	
	100001	01	SCIF_1 transmit	TXI1 (transmit FIFO data empty)	Any	SCFTDR_1	
		10	SCIF_1 receive	RXI1 (receive FIFO data full)	SCFRDR_1	Any	
	100010	01	SCIF_2 transmit	TXI2 (transmit FIFO data empty)	Any	SCFTDR_2	
		10	SCIF_2 receive	RXI2 (receive FIFO data full)	SCFRDR_2	Any	
	100011	01	SCIF_3 transmit	TXI3 (transmit FIFO data empty)	Any	SCFTDR_3	
		10	SCIF_3 receive	RXI3 (receive FIFO data full)	SCFRDR_3	Any	
	101000	01	IIC3 transmit	TXI (transmit data empty)	Any	ICDRT	Cycle steal
		10	IIC3 receive	RXI (receive data full)	ICDRR	Any	
	101100	11	A/D converter	ADI (A/D conversion end)	ADDR	Any	Cycle steal
	111000	11	MTU2_0	TGI0A	Any	Any	Cycle steal or burst
	111001	11	MTU2_1	TGI1A	Any	Any	
	111010	11	MTU2_2	TGI2A	Any	Any	
	111011	11	MTU2_3	TGI3A	Any	Any	
	111100	11	MTU2_4	TGI4A	Any	Any	
	111110	11	CMT_0	Compare match 0	Any	Any	Cycle steal or burst
	111111	11	CMT_1	Compare match 1	Any	Any	

9.4.3 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 9.3. 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.

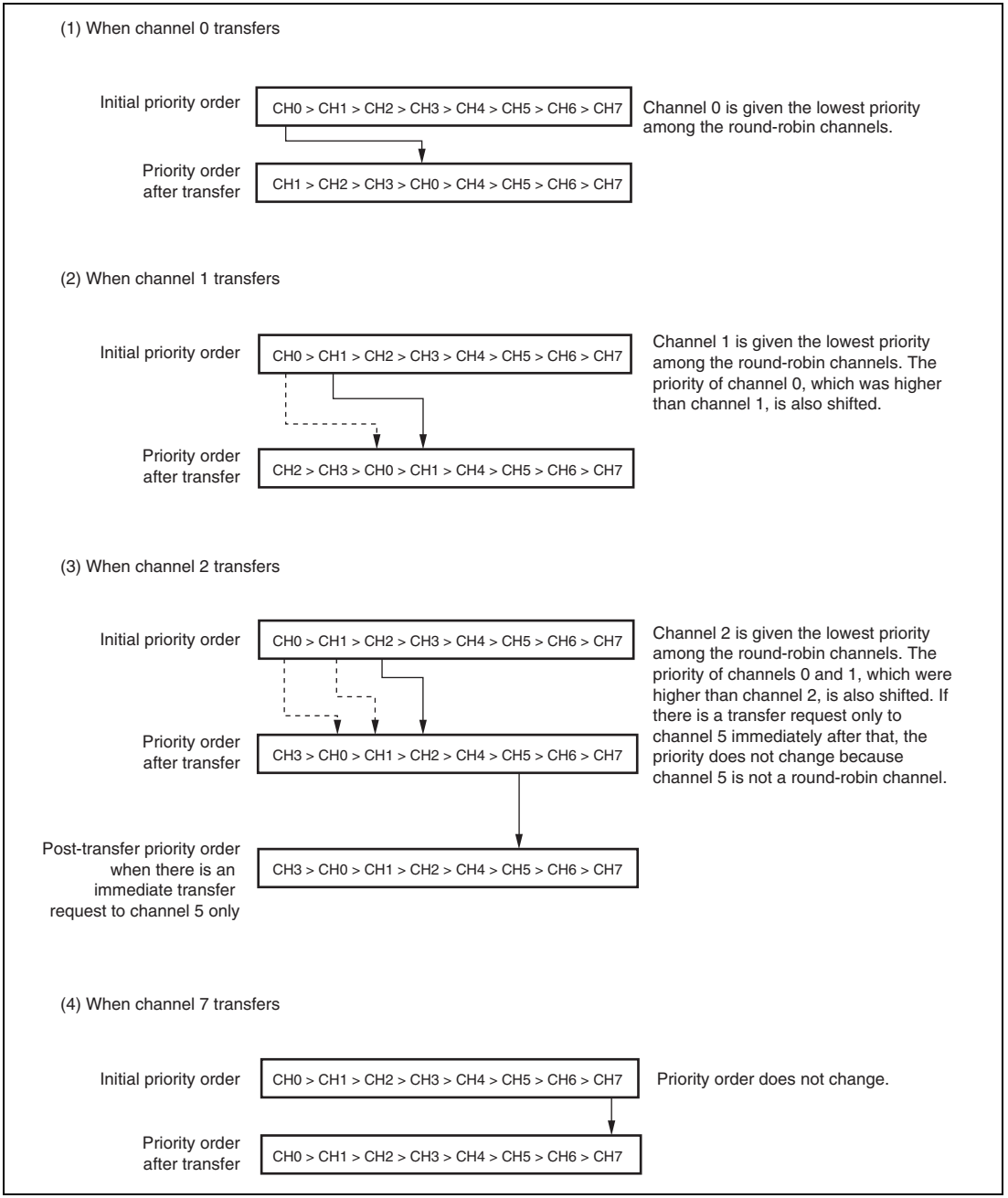


Figure 9.3 Round-Robin Mode

Figure 9.4 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.

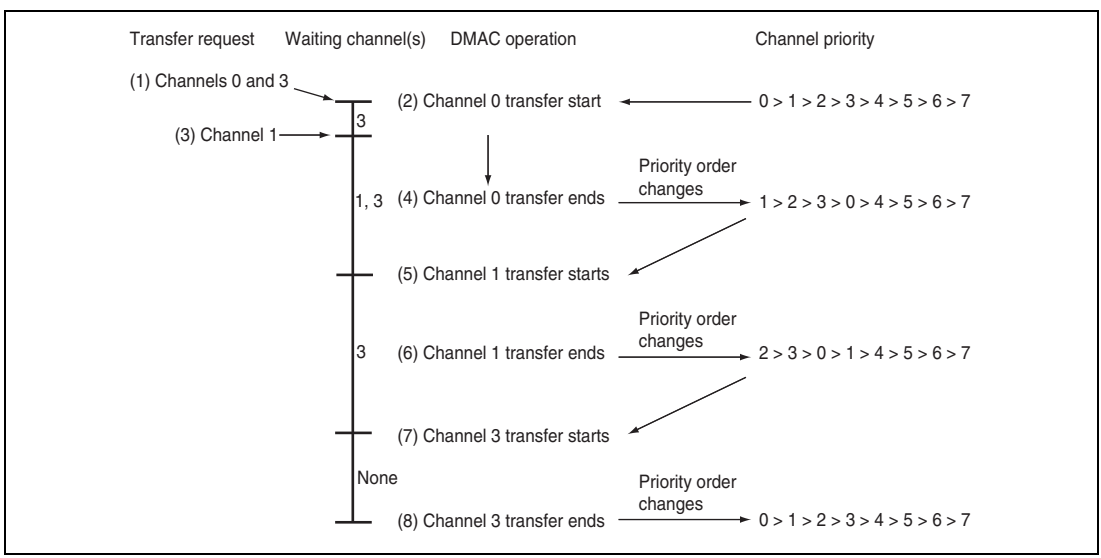


Figure 9.4 Changes in Channel Priority in Round-Robin Mode

9.4.4 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 9.9.

Table 9.9 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 9.5, 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.

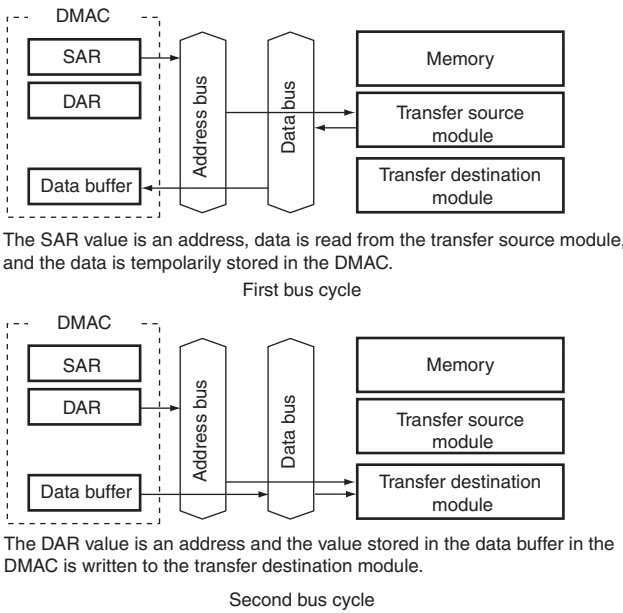
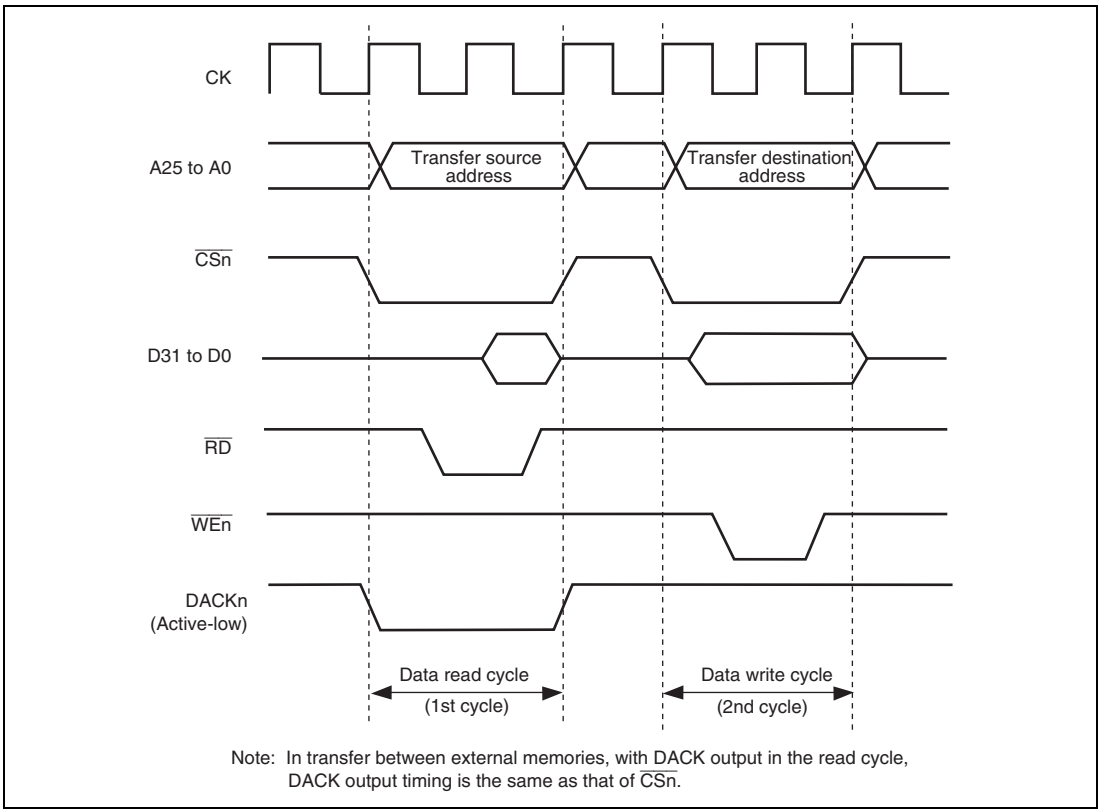


Figure 9.5 Data Flow of Dual Address Mode

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 9.6 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 9.6 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 9.7, when the external device outputs data to the data bus, that data is written to the external memory in the same bus cycle.

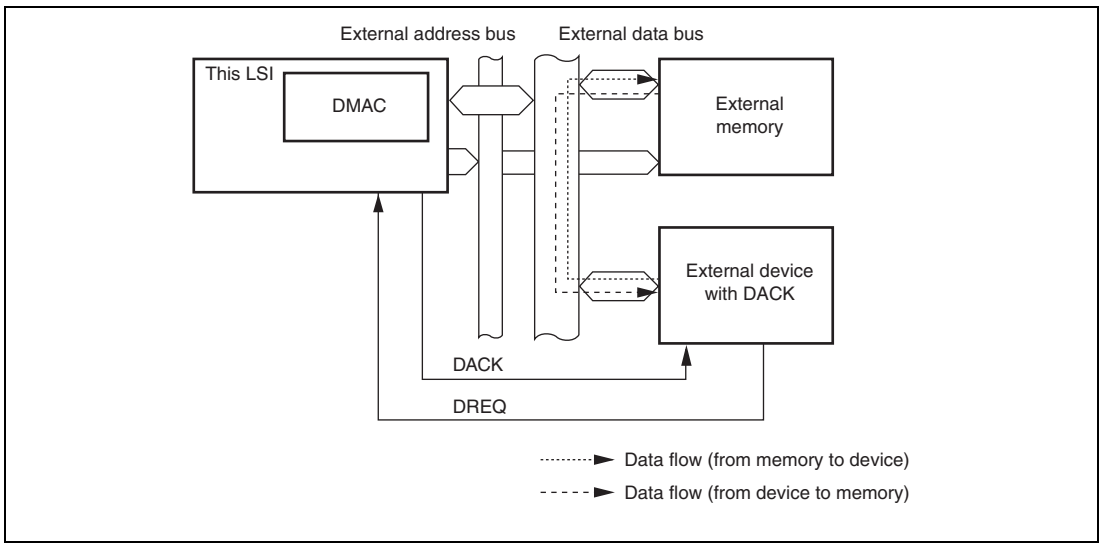


Figure 9.7 Data Flow in Single Address Mode

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 9.8 shows an example of DMA transfer timing in single address mode.

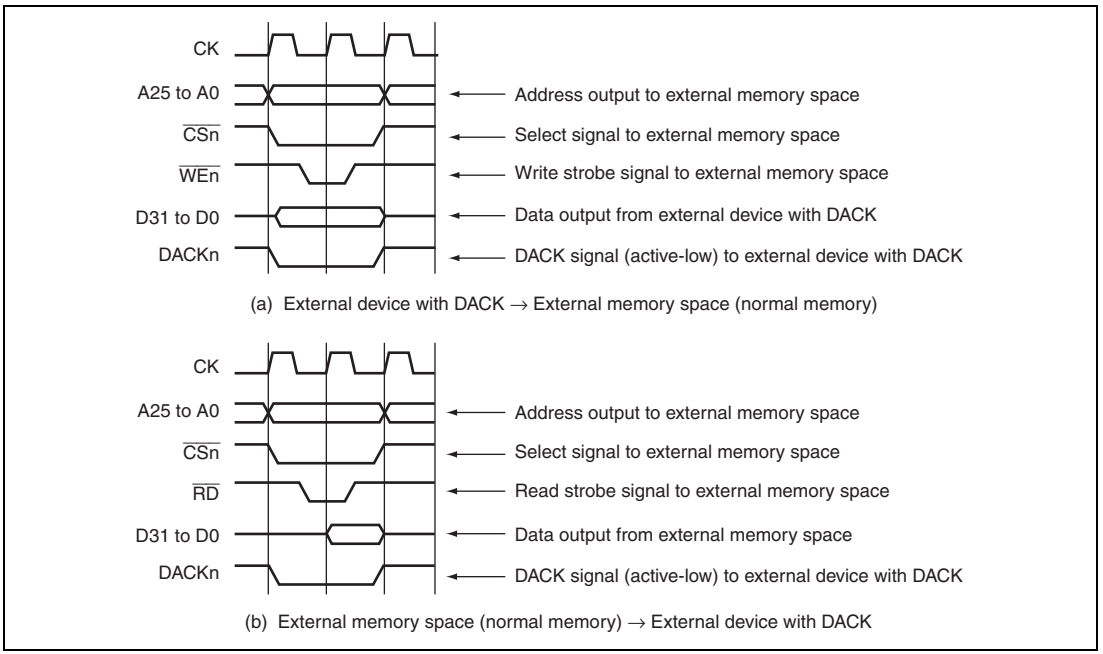


Figure 9.8 Example of DMA Transfer Timing in Single Address Mode

(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 9.9 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

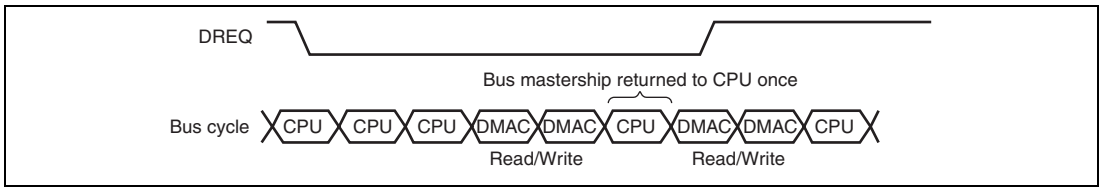


Figure 9.9 DMA Transfer Example in Cycle-Steal Normal Mode (Dual Address, DREQ Low Level Detection)

- 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 9.10 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

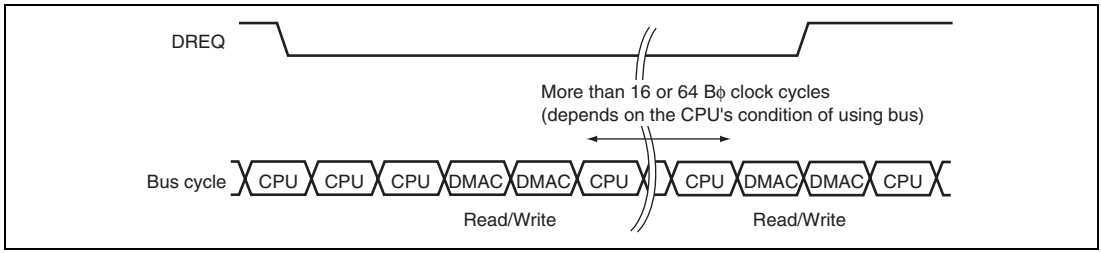


Figure 9.10 Example of DMA Transfer in Cycle-Steal Intermittent Mode (Dual Address, DREQ Low Level Detection)

(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 9.11 shows DMA transfer timing in burst mode.

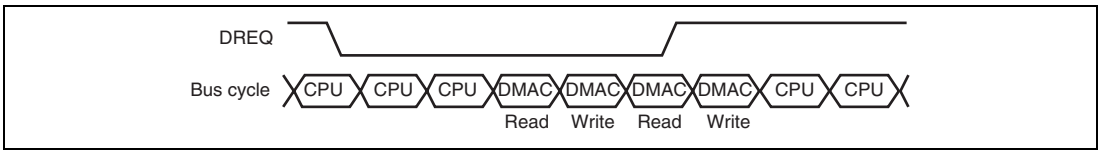


Figure 9.11 DMA Transfer Example in Burst Mode (Dual Address, DREQ Low Level Detection)

(3) Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 9.10 shows the relationship between request modes and bus modes by DMA transfer category.

Table 9.10 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 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 9.12.

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.

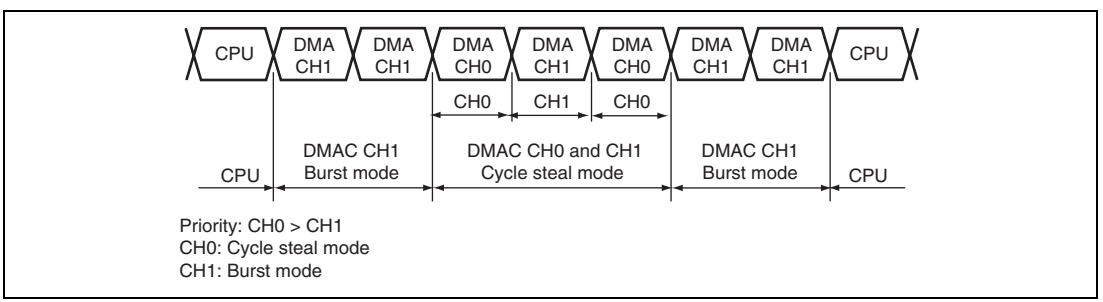


Figure 9.12 Bus State when Multiple Channels are Operating

In round-robin mode, the priority changes as shown in figure 9.3. Note that channels in cycle steal and burst modes must not be mixed.

9.4.5 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 8, Bus State Controller (BSC).

(2) DREQ Pin Sampling Timing

Figures 9.13 to 9.16 show the DREQ input sampling timings in each bus mode.

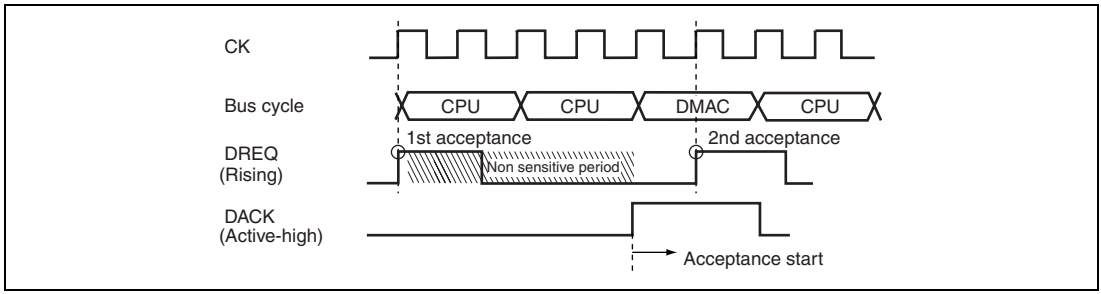


Figure 9.13 Example of DREQ Input Detection in Cycle Steal Mode Edge Detection

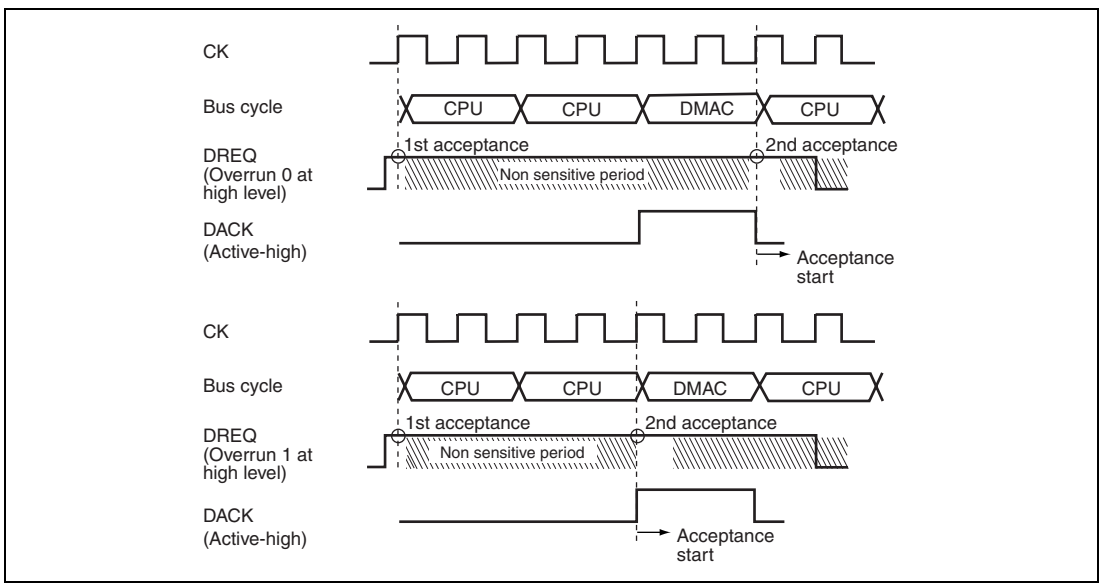


Figure 9.14 Example of DREQ Input Detection in Cycle Steal Mode Level Detection

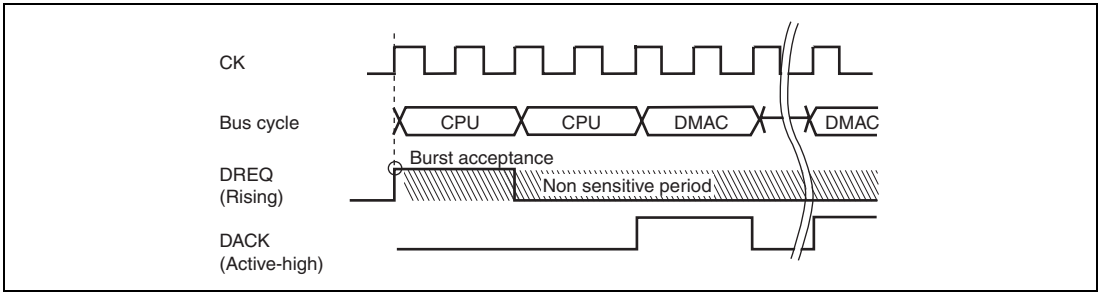


Figure 9.15 Example of DREQ Input Detection in Burst Mode Edge Detection

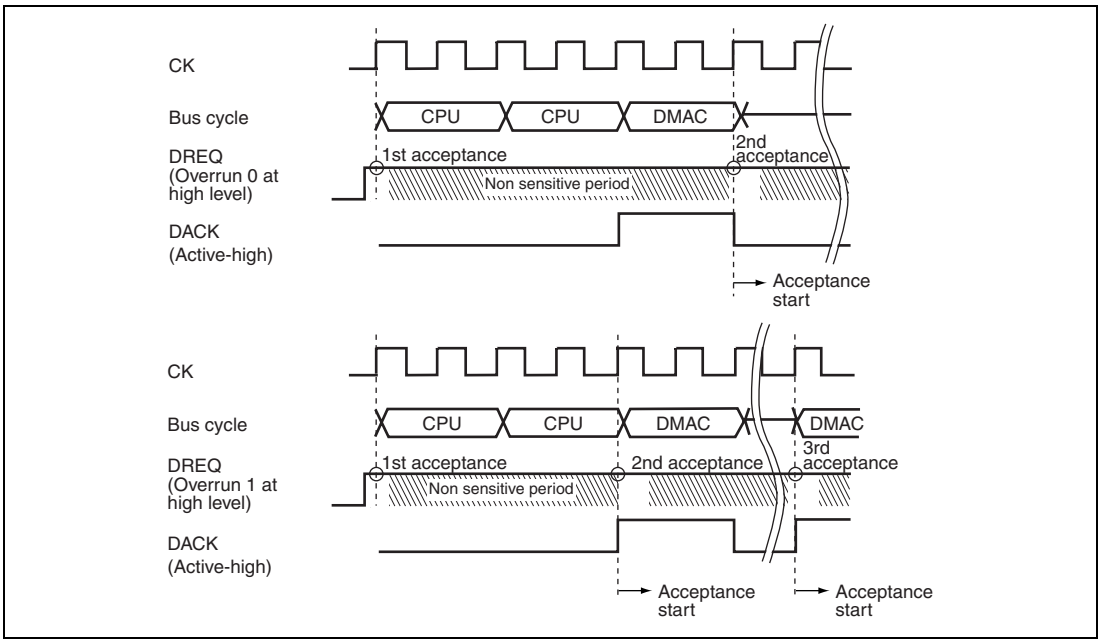


Figure 9.16 Example of DREQ Input Detection in Burst Mode Level Detection

Figure 9.17 shows the TEND output timing.

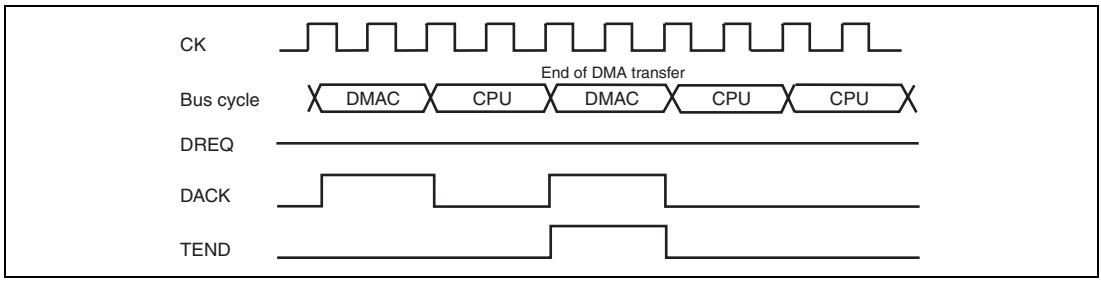
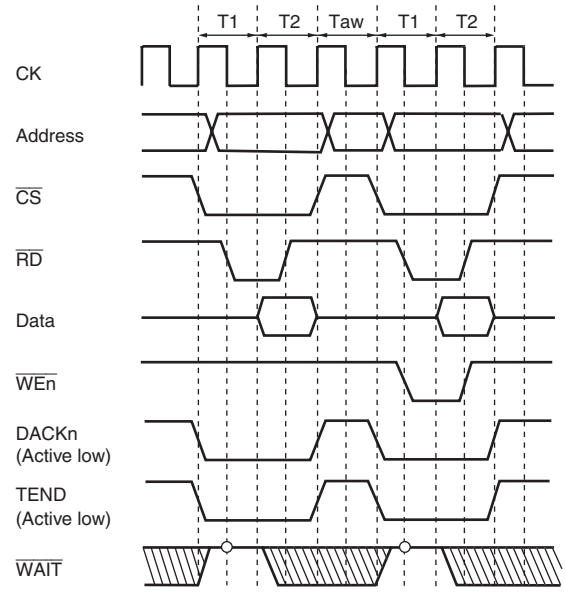


Figure 9.17 Example of DMA Transfer End Signal Timing (Cycle Steal Mode Level Detection)

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 9.18 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 $\overline{CS}$ is negated between the bus cycles, TEND is also divided.

Figure 9.18 BSC Normal Memory Access (No Wait, Idle Cycle 1, Longword Access to 16-Bit Device)

9.5 Usage Note

9.5.1 Half-End Flag Setting and Half-End Interrupt

When monitoring the half-end flag status in CHCR or using the half-end interrupt together with the reload function, the following precautions must be observed.

For the reload transfer count in RDMATCR, always set a value equal to the initial transfer count (the value in DMATCR).

If the first setting of DMATCR differs from the RDMATCR setting used in the second and following DMA transfer, the half-end flag setting timing may be earlier than half of the transfer count or the half-end flag may not be set. The same is true for the half-end interrupt.

Section 10 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.

10.1 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. However, waveform output by compare match for channel 5 is not possible.
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- Dead time compensation counter available in channel 5
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

Table 10.1 MTU2 Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64
General registers		TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRU_5 TGRV_5 TGRW_5
General registers/ buffer registers		TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4	—
I/O pins		TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	Input pins TIC5U TIC5V TIC5W
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	√	√	√	√	√	—
	1 output	√	√	√	√	√	—
	Toggle output	√	√	√	√	√	—
Input capture function		√	√	√	√	√	√
Synchronous operation		√	√	√	√	√	—
PWM mode 1		√	√	√	√	√	—
PWM mode 2		√	√	√	—	—	—
Complementary PWM mode		—	—	—	√	√	—
Reset PWM mode		—	—	—	√	√	—
AC synchronous motor drive mode		√	—	—	√	√	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Phase counting mode	—	√	√	—	—	—
Buffer operation	√	—	—	√	√	—
Dead time compensation counter function	—	—	—	—	—	√
DMAC activation	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	—
A/D converter start trigger	TGRA_0 compare match or input capture TGRE_0 compare match	TGRA_1 compare match or input capture	TGRA_2 compare match or input capture	TGRA_3 compare match or input capture	TGRA_4 compare match or input capture TCNT_4 underflow (trough) in complementary PWM mode	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Interrupt sources	7 sources	4 sources	4 sources	5 sources	5 sources	3 sources
	• Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	• Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	• Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	• Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	• Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow	• Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W

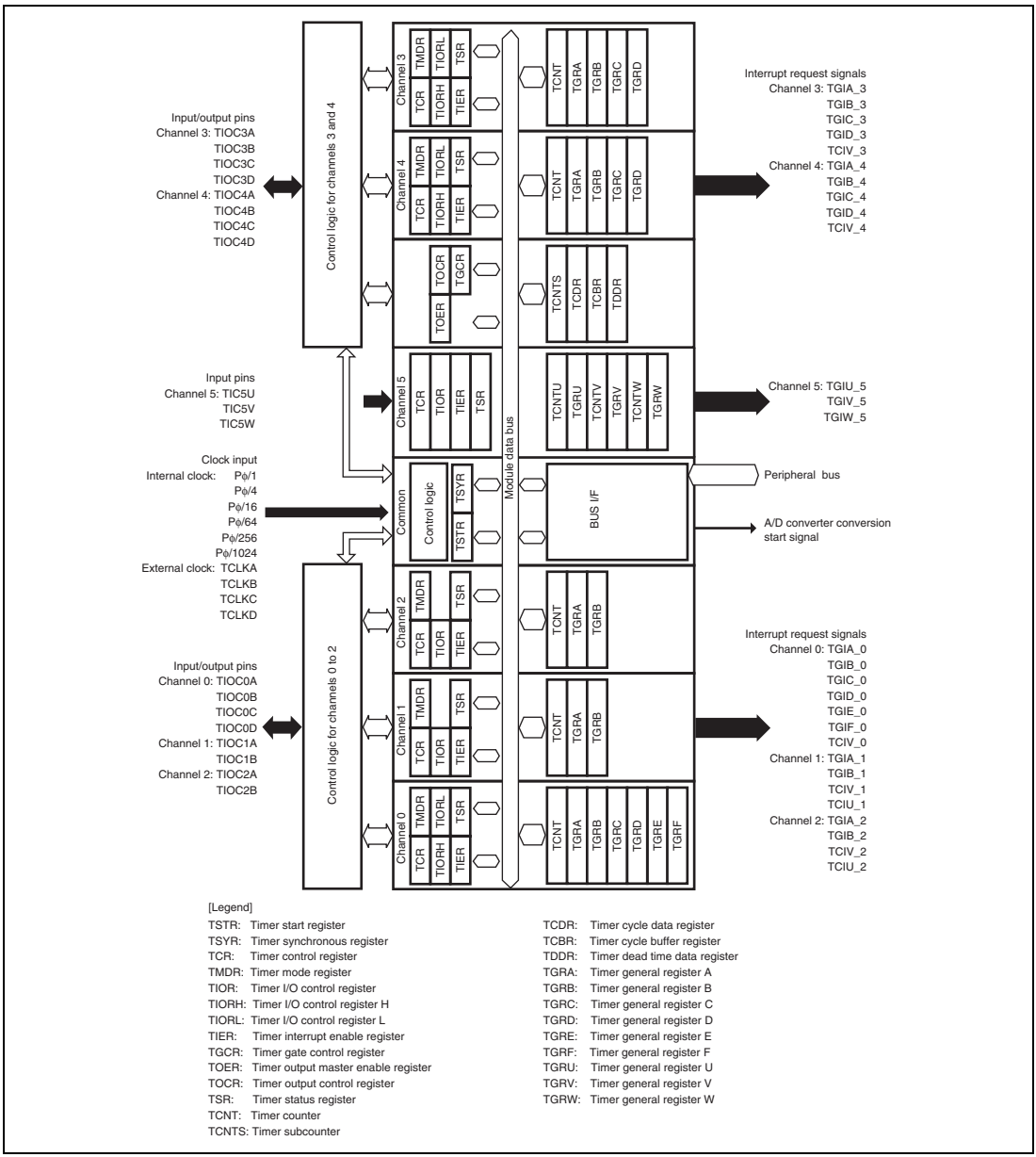
Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA_4 and TCNT_4 • A/D converter start request at a match between TADCORB_4 and TCNT_4	—
Interrupt skipping function	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts	—

[Legend]

√: Possible

—: Not possible

Figure 10.1 shows a block diagram of the MTU2.



[Legend]

TSTR: Timer start register
TSYR: Timer synchronous register
TCR: Timer control register
TMDR: Timer mode register
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TGCR: Timer gate control register
TOER: Timer output master enable register
TOCR: Timer output control register
TSR: Timer status register
TCNT: Timer counter
TCNTS: Timer subcounter
TCDR: Timer cycle data register
TCBR: Timer cycle buffer register
TDDR: Timer dead time data register
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TGRU: Timer general register U
TGRV: Timer general register V
TGRW: Timer general register W

Figure 10.1 Block Diagram of MTU2

10.2 Input/Output Pins

Table 10.2 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

Note: For the pin configuration in complementary PWM mode, see table 10.54 in section 10.4.8, Complementary PWM Mode.

10.3 Register Descriptions

The MTU2 has the following registers. For details on register addresses and register states during each process, refer to section 26, 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 10.3 Register Descriptions

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
0	Timer control register_0	TCR_0	R/W	H'00	H'FFFE4300	8
	Timer mode register_0	TMDR_0	R/W	H'00	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	R/W	H'00	H'FFFE4302	8
	Timer I/O control register L_0	TIORL_0	R/W	H'00	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	R/W	H'00	H'FFFE4304	8
	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
	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
	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
	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
	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
1	Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380	8
	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
	Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385	8

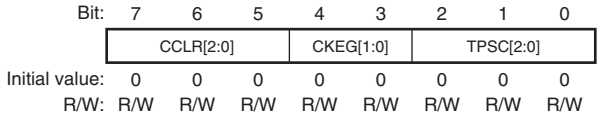
Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
1	Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE4386	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16
	Timer general register B_1	TGRB_1	R/W	H'FFFF	H'FFFE438A	16
	Timer input capture control register	TICCR	R/W	H'00	H'FFFE4390	8
2	Timer control register_2	TCR_2	R/W	H'00	H'FFFE4000	8
	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
	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
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16
3	Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200	8
	Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202	8
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204	8
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208	8
	Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C	8
	Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210	16
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238	8
4	Timer control register_4	TCR_4	R/W	H'00	H'FFFE4201	8
	Timer mode register_4	TMDR_4	R/W	H'00	H'FFFE4203	8
	Timer I/O control register H_4	TIORH_4	R/W	H'00	H'FFFE4206	8
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
4	Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE4209	8
	Timer status register_4	TSR_4	R/W	H'C0	H'FFFE422D	8
	Timer counter_4	TCNT_4	R/W	H'0000	H'FFFE4212	16
	Timer general register A_4	TGRA_4	R/W	H'FFFF	H'FFFE421C	16
	Timer general register B_4	TGRB_4	R/W	H'FFFF	H'FFFE421E	16
	Timer general register C_4	TGRC_4	R/W	H'FFFF	H'FFFE4228	16
	Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	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
	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
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16
5	Timer control register U_5	TCRU_5	R/W	H'00	H'FFFE4084	8
	Timer control register V_5	TCRV_5	R/W	H'00	H'FFFE4094	8
	Timer control register W_5	TCRW_5	R/W	H'00	H'FFFE40A4	8
	Timer I/O control register U_5	TIORU_5	R/W	H'00	H'FFFE4086	8
	Timer I/O control register V_5	TIORV_5	R/W	H'00	H'FFFE4096	8
	Timer I/O control register W_5	TIORW_5	R/W	H'00	H'FFFE40A6	8
	Timer interrupt enable register_5	TIER_5	R/W	H'00	H'FFFE40B2	8
	Timer status register_5	TSR_5	R/W	H'00	H'FFFE40B0	8
	Timer start register_5	TSTR_5	R/W	H'00	H'FFFE40B4	8
	Timer counter U_5	TCNTU_5	R/W	H'0000	H'FFFE4080	16
	Timer counter V_5	TCNTV_5	R/W	H'0000	H'FFFE4090	16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
5	Timer counter W_5	TCNTW_5	R/W	H'0000	H'FFFE40A0	16
	Timer general register U_5	TGRU_5	R/W	H'FFFF	H'FFFE4082	16
	Timer general register V_5	TGRV_5	R/W	H'FFFF	H'FFFE4092	16
	Timer general register W_5	TGRW_5	R/W	H'FFFF	H'FFFE40A2	16
	Timer compare match clear register	TCNTCMPCLR	R/W	H'00	H'FFFE40B6	8
Common	Timer start register	TSTR	R/W	H'00	H'FFFE4280	8
	Timer synchronous register	TSYR	R/W	H'00	H'FFFE4281	8
	Timer counter synchronous start register	TCSYSTR	R/W	H'00	H'FFFE4282	8
	Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284	8
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A	8
	Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E	8
	Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F	8
	Timer gate control register	TGCR	R/W	H80	H'FFFE420D	8
	Timer cycle control register	TCDR	R/W	H'FFFF	H'FFFE4214	16
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216	16
	Timer subcounter	TCNTS	R	H'0000	H'FFFE4220	16
	Timer cycle buffer register	TCBR	R/W	H'FFFF	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	R/W	H'00	H'FFFE4230	8
	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 synchronous clear register	TSYCR	R/W	H'00	H'FFFE4250	8
	Timer waveform control register	TWCR	R/W	H'00	H'FFFE4260	8
	Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236	8

10.3.1 Timer Control Register (TCR)

The TCR registers are 8-bit readable/writable registers that control the TCNT operation for each channel. The MTU2 has a total of eight TCR registers, one each for channels 0 to 4 and three (TCRU_5, TCRV_5, and TCRW_5) for channel 5. TCR register settings should be conducted only when TCNT operation is stopped.



Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See tables 10.4 and 10.5 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. Pϕ/4 both edges = Pϕ/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 Pϕ/4 or slower. When Pϕ/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 10.6 to 10.10 for details.

[Legend]
x: Don't care

Table 10.4 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 10.5 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 10.6 TPSC0 to TPSC2 (Channel 0)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

Table 10.7 TPSC0 to TPSC2 (Channel 1)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on Pϕ/256
			1	Counts on TCNT_2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Table 10.8 TPSC0 to TPSC2 (Channel 2)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on Pϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Table 10.9 TPSC0 to TPSC2 (Channels 3 and 4)

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3, 4	0	0	0	Internal clock: counts on Pϕ/1
			1	Internal clock: counts on Pϕ/4
		1	0	Internal clock: counts on Pϕ/16
			1	Internal clock: counts on Pϕ/64
	1	0	0	Internal clock: counts on Pϕ/256
			1	Internal clock: counts on Pϕ/1024
		1	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input

Table 10.10 TPSC1 and TPSC0 (Channel 5)

Channel	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	Internal clock: counts on Pϕ/1
		1	Internal clock: counts on Pϕ/4
	1	0	Internal clock: counts on Pϕ/16
		1	Internal clock: counts on Pϕ/64

Note: Bits 7 to 2 are reserved in channel 5. These bits are always read as 0. The write value should always be 0.

10.3.2 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 10.11 for details.

Table 10.11 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.

10.3.3 Timer I/O Control Register (TIOR)

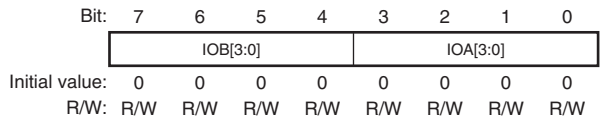
The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU2 has a total of eleven TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2, and three (TIORU_5, TIORV_5, and TIORW_5) for channel 5.

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.

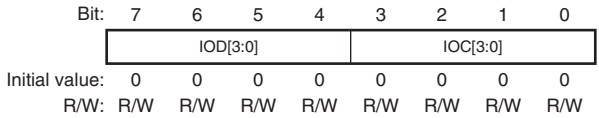
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

- TIORH_0, TIOR_1, TIOR_2, TIORH_3, TIORH_4



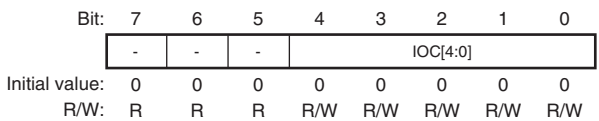
Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOB[3:0]	0000	R/W	I/O Control B0 to B3 Specify the function of TGRB. See the following tables. TIORH_0: Table 10.12 TIOR_1: Table 10.14 TIOR_2: Table 10.15 TIORH_3: Table 10.16 TIORH_4: Table 10.18
3 to 0	IOA[3:0]	0000	R/W	I/O Control A0 to A3 Specify the function of TGRA. See the following tables. TIORH_0: Table 10.20 TIOR_1: Table 10.22 TIOR_2: Table 10.23 TIORH_3: Table 10.24 TIORH_4: Table 10.26

- 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 10.13 TIORL_3: Table 10.17 TIORL_4: Table 10.19
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 10.21 TIORL_3: Table 10.25 TIORL_4: Table 10.27

- 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 10.28.

Table 10.12 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 10.13 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 10.14 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 10.15 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 10.16 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 10.17 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 10.18 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 10.19 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 10.20 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 10.21 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 10.22 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 10.23 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 10.24 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 10.25 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 10.26 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 10.27 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 10.28 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

10.3.4 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

10.3.5 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 this 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 this 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 this bit in TSR_5 is set to 1. 0: Interrupt requests (TGIW_5) disabled 1: Interrupt requests (TGIW_5) enabled

10.3.6 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 [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 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 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 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. When writing to the timer status register (TSR), write 0 to the bit to be cleared after reading 1. Write 1 to other bits. But 1 is not actually written and the previous value is held.

- 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. When writing to the timer status register (TSR), write 0 to the bit to be cleared after reading 1. Write 1 to other bits. But 1 is not actually written and the previous value is held.

- 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 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 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 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).

10.3.7 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

10.3.8 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

10.3.9 Timer Synchronous Clear Register (TSYCR)

TSYCR 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 TSYCR in channel 3 but the MTU2 has no TSYCR.

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

10.3.10 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 10.29.
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 10.29 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.

10.3.11 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.

10.3.12 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.

10.3.13 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.

10.3.14 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.

10.3.15 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

10.3.16 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	

10.3.17 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.

10.3.18 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.

10.3.19 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 10.3.20, Timer Output Control Register 1 (TOCR1), and section 10.3.21, 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.

10.3.20 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)*1	TOC Register Write Protection*2 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*3 This bit selects the reverse phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 10.30.
0	OLSP	0	R/W	Output Level Select P*3 This bit selects the positive phase output level in reset-synchronized PWM mode/complementary PWM mode. See table 10.31.

Notes: 1. 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.
2. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
3. Clearing the TOCS0 bit to 0 makes this bit setting valid.

Table 10.30 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 10.31 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 10.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1, OLSP = 1.

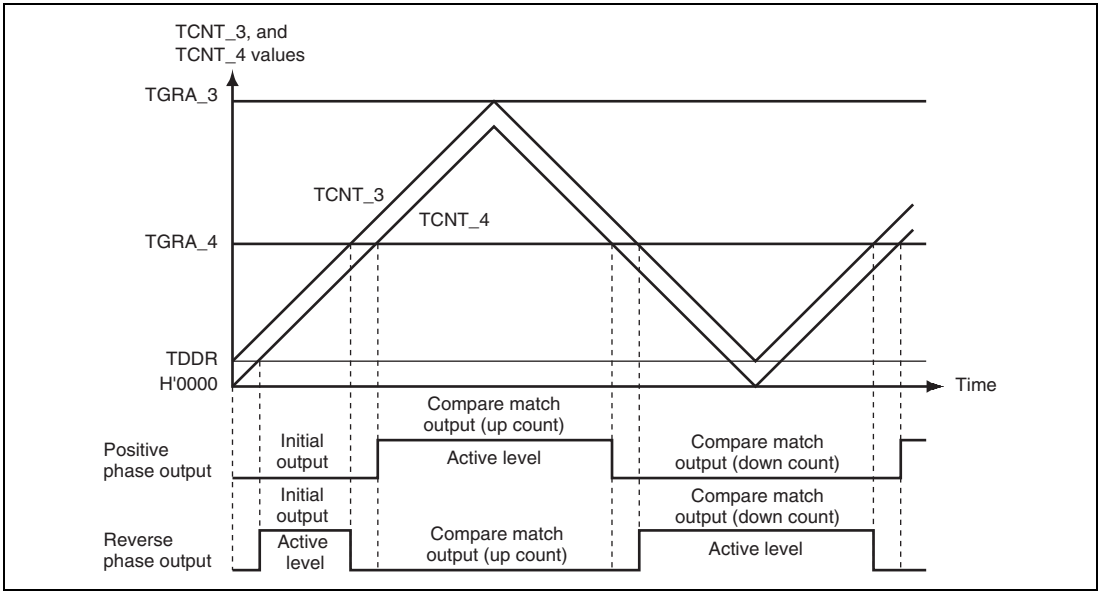


Figure 10.2 Complementary PWM Mode Output Level Example

10.3.21 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 10.32.
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 10.33.
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 10.34.
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 10.35.
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 10.36.
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 10.37.

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 10.38.

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

Table 10.32 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 10.33 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 10.34 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 10.35 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 10.36 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 10.37 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 10.38 TIOC4B 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

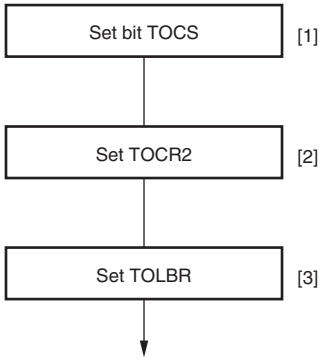
10.3.22 Timer Output Level Buffer Register (TOLBR)

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 10.3 shows an example of the PWM output level setting procedure in buffer operation.



[1] Set bit TOCS in TOCR1 to 1 to enable the TOCR2 setting.

[2] Use bits BF1 and BF0 in TOCR2 to select the TOLBR buffer transfer timing. Use bits OLS3N to OLS1N and OLS3P to OLS1P to specify the PWM output levels.

[3] The TOLBR initial setting must be the same value as specified in bits OLS3N to OLS1N and OLS3P to OLS1P in TOCR2.

Figure 10.3 PWM Output Level Setting Procedure in Buffer Operation

10.3.23 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 10.39.
1	VF	0	R/W	
0	UF	0	R/W	

Note: * If the BDC bit in the MTU2S is set to 1, the FB bit should not be cleared to 0.

Table 10.39 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

10.3.24 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.

10.3.25 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.

10.3.26 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.

10.3.27 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.

10.3.28 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 10.40.
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 10.41.

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 10.40 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 10.41 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.

10.3.29 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.

10.3.30 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 10.42.

Note: * Applicable buffer registers:
TGRC_3, TGRD_3, TGRC_4, TGRD_4, and TCBR

Table 10.42 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 10.4.8, 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.

10.3.31 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.

10.3.32 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 10.40. 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)	Waveform Retain Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The output waveform is retained 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 10.40. 0: Outputs the initial value specified in TOCR 1: Retains the waveform output immediately before synchronous clearing [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.

10.3.33 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.

10.4 Operation

10.4.1 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 10.4 shows an example of the count operation setting procedure.

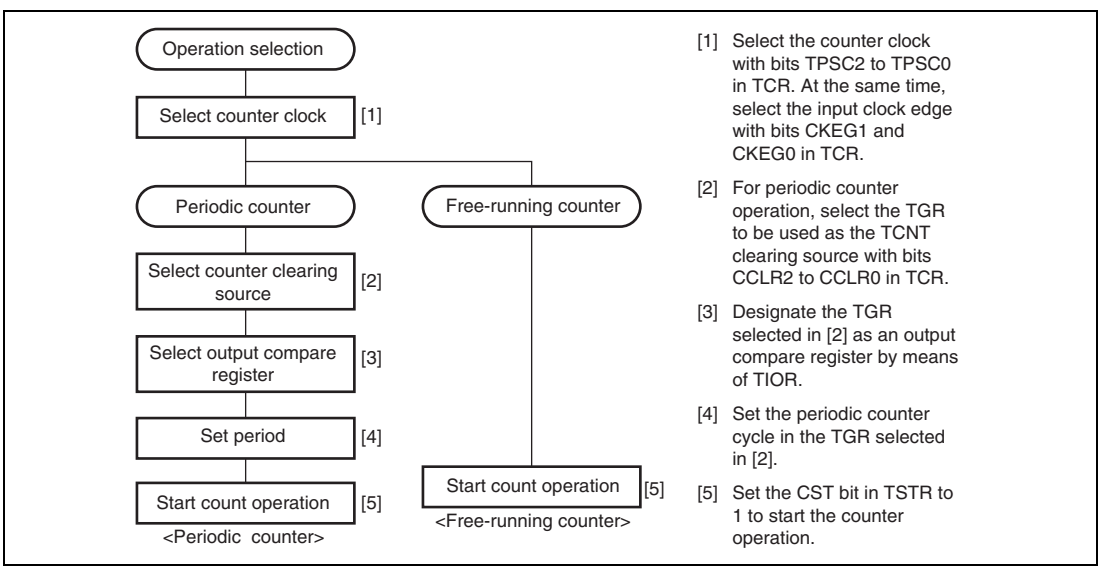


Figure 10.4 Example of Counter Operation Setting Procedure

(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 10.5 illustrates free-running counter operation.

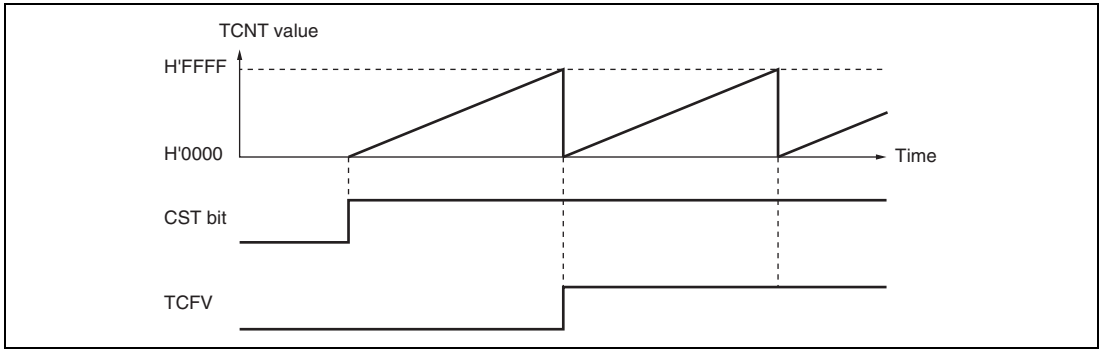


Figure 10.5 Free-Running Counter Operation

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 10.6 illustrates periodic counter operation.

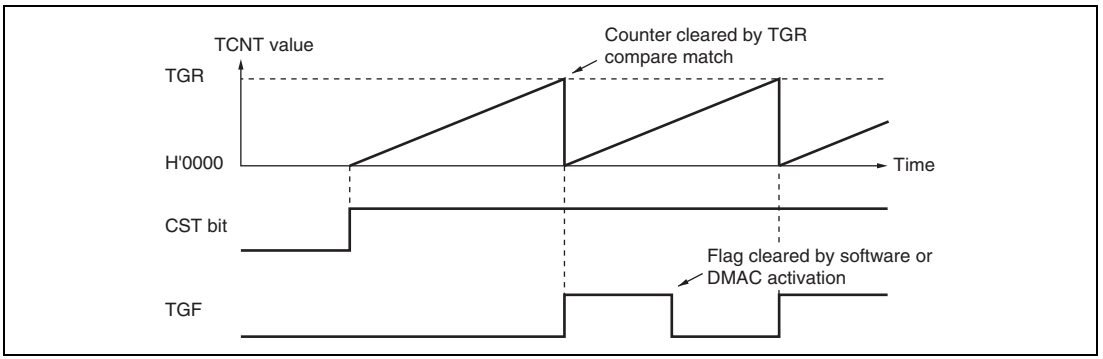


Figure 10.6 Periodic Counter Operation

(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 10.7 shows an example of the setting procedure for waveform output by compare match

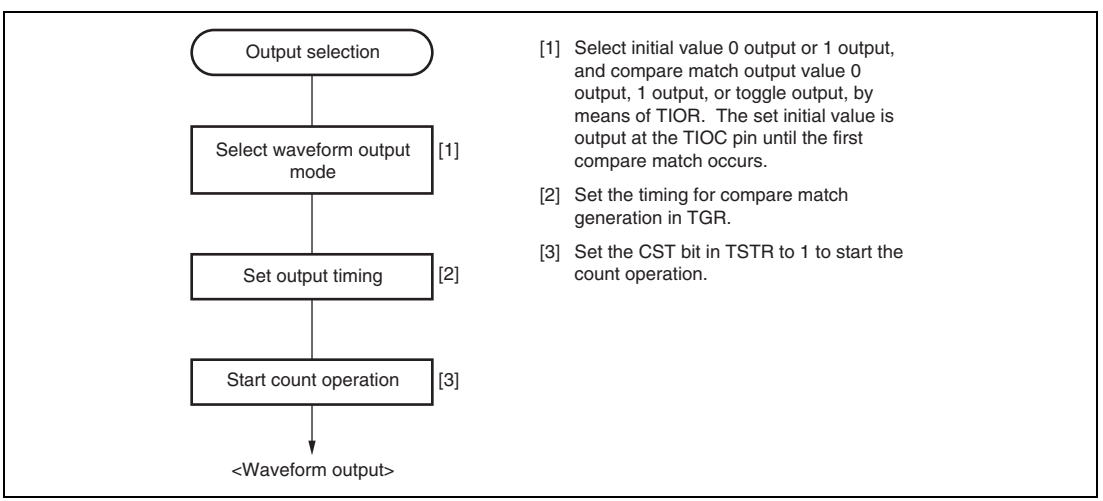


Figure 10.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 10.8 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.

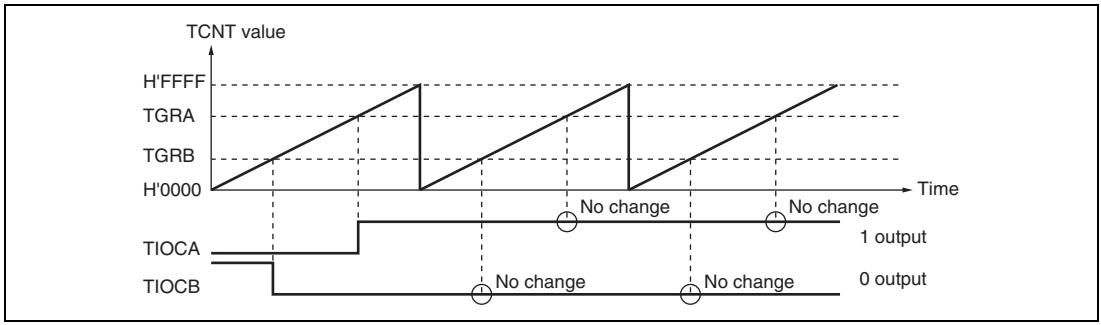


Figure 10.8 Example of 0 Output/1 Output Operation

Figure 10.9 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.

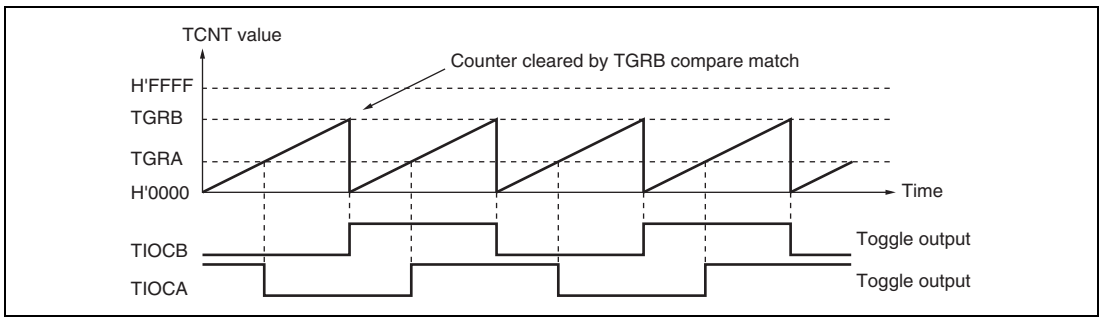


Figure 10.9 Example of Toggle Output Operation

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0 and 1, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 1, Pϕ/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if Pϕ/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 10.10 shows an example of the input capture operation setting procedure.

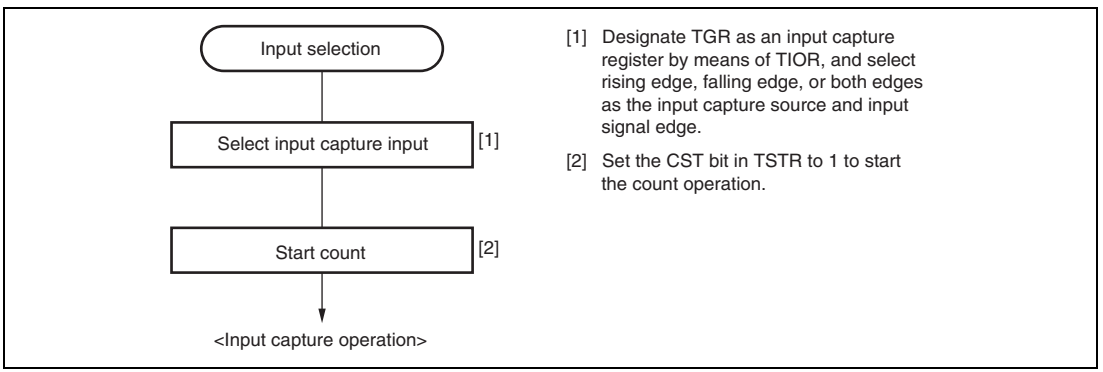


Figure 10.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 10.11 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.

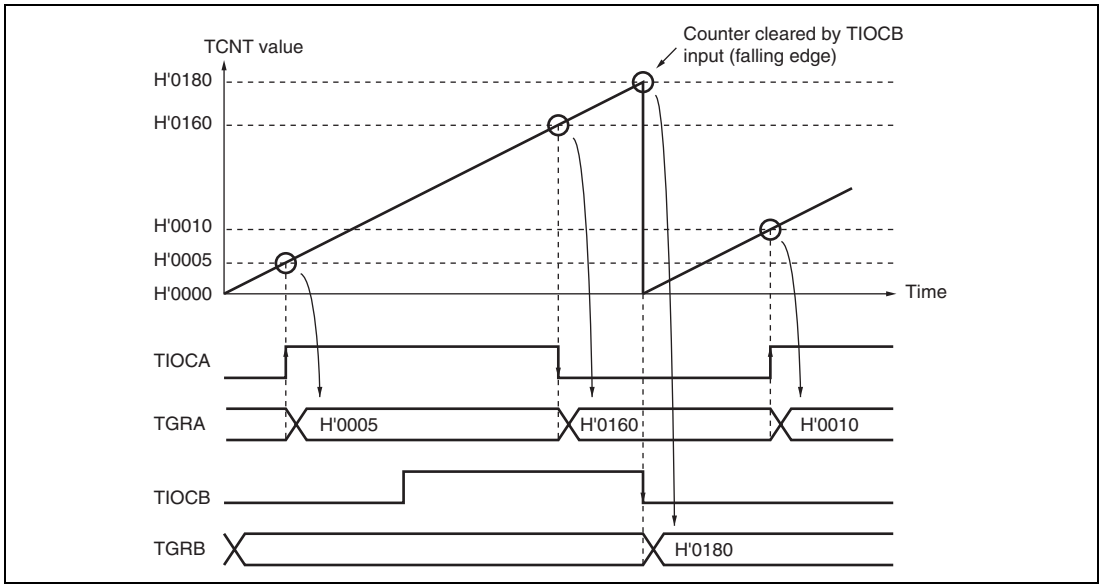


Figure 10.11 Example of Input Capture Operation

10.4.2 Synchronous Operation

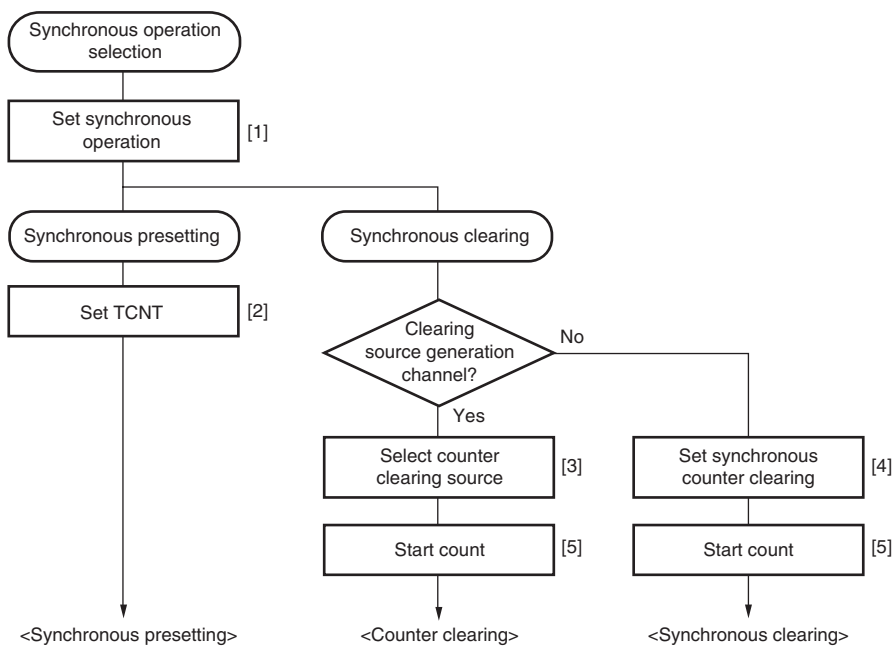
In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

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 10.12 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 10.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 10.13 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 10.4.5, PWM Modes.

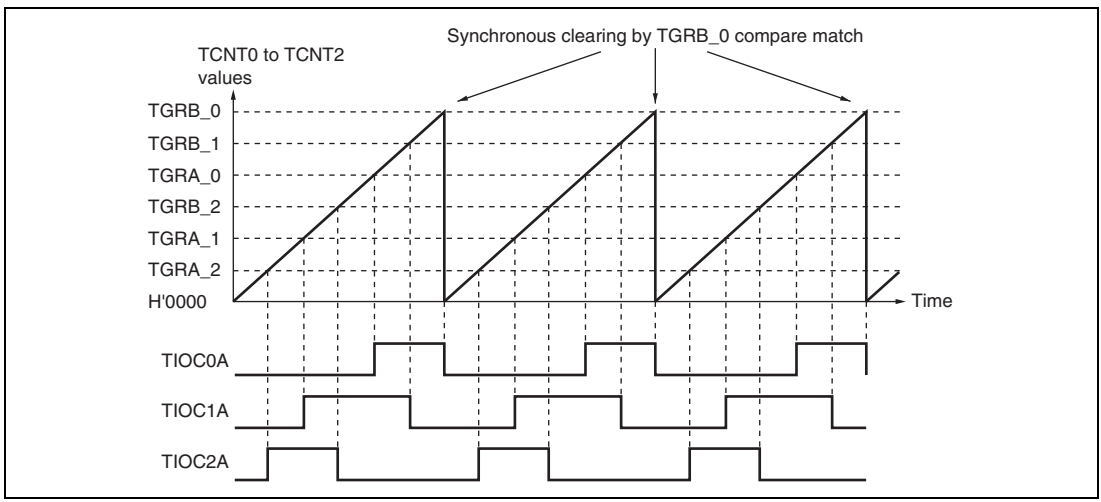


Figure 10.13 Example of Synchronous Operation

10.4.3 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 10.43 shows the register combinations used in buffer operation.

Table 10.43 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 10.14.

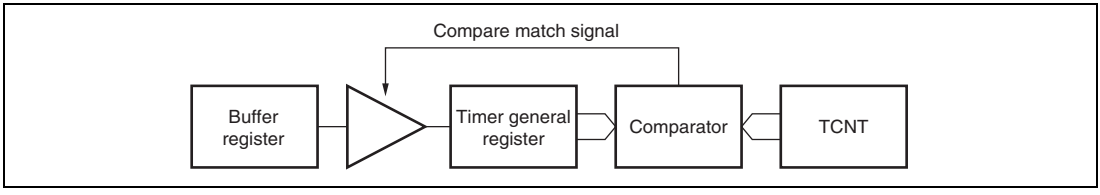


Figure 10.14 Compare Match Buffer Operation

- 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 10.15.

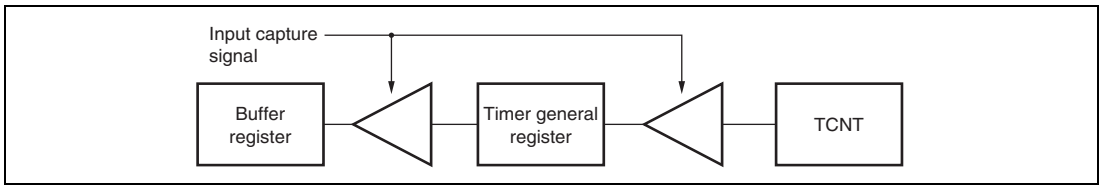


Figure 10.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 10.16 shows an example of the buffer operation setting procedure.

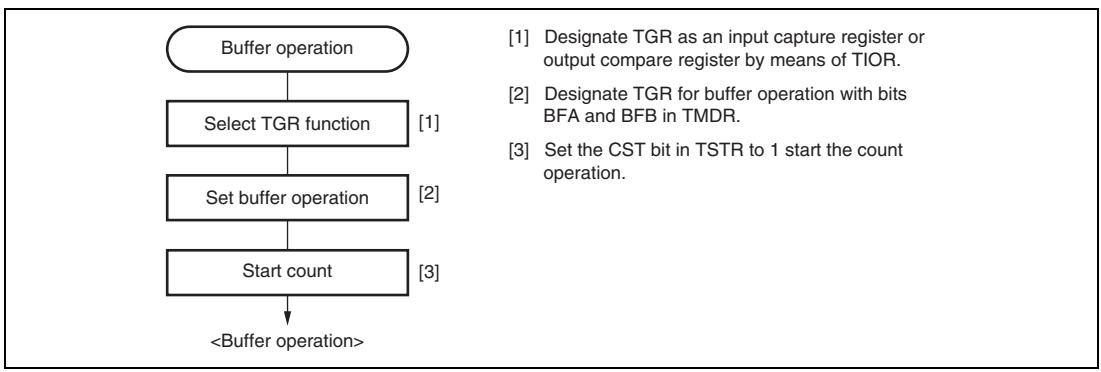


Figure 10.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 10.17 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 10.4.5, PWM Modes.

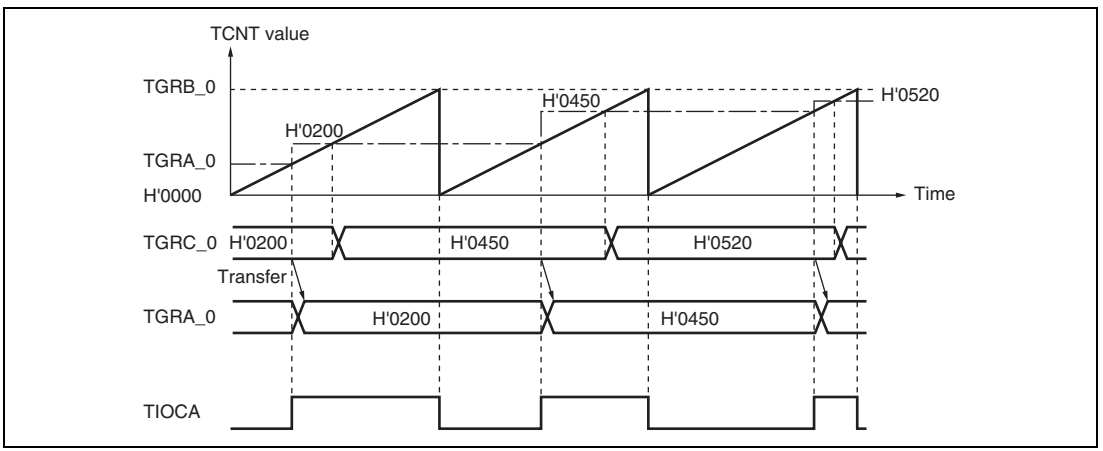


Figure 10.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 10.18 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.

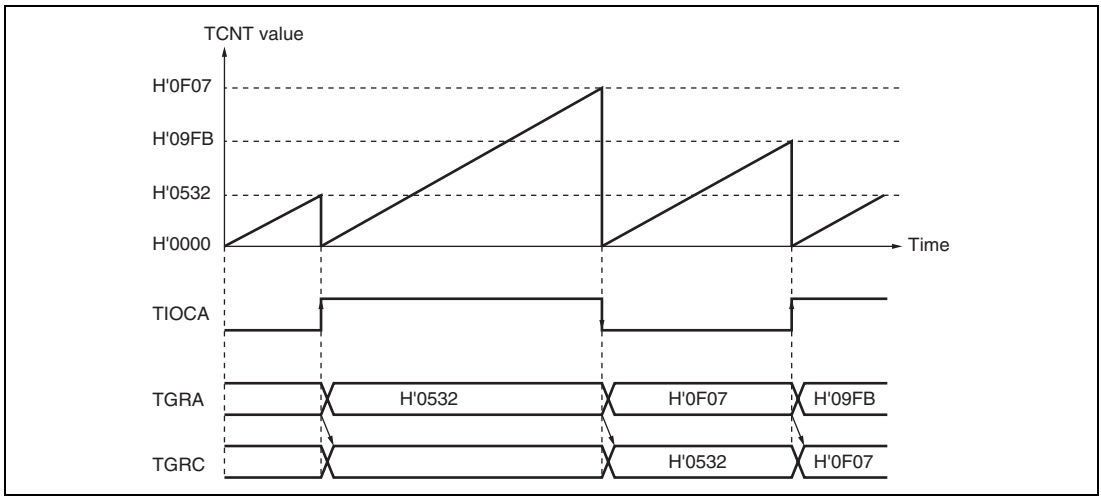


Figure 10.18 Example of Buffer Operation (2)

(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 10.19 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.

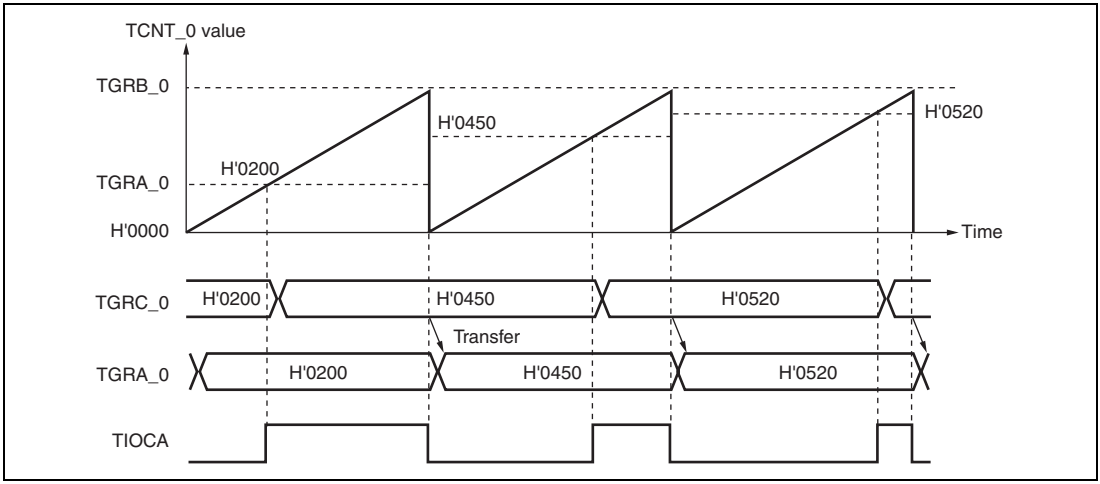


Figure 10.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

10.4.4 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 10.44 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 10.44 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 10.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 10.45 show the TICCR setting and input capture input pins.

Table 10.45 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 10.20 shows an example of the setting procedure for cascaded operation.

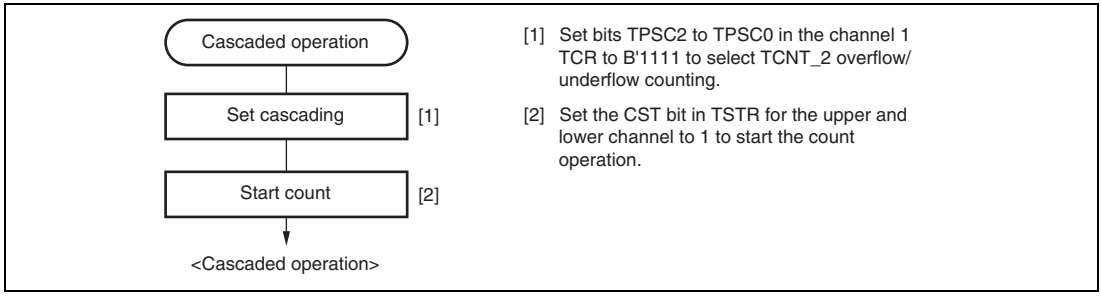


Figure 10.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 10.21 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.

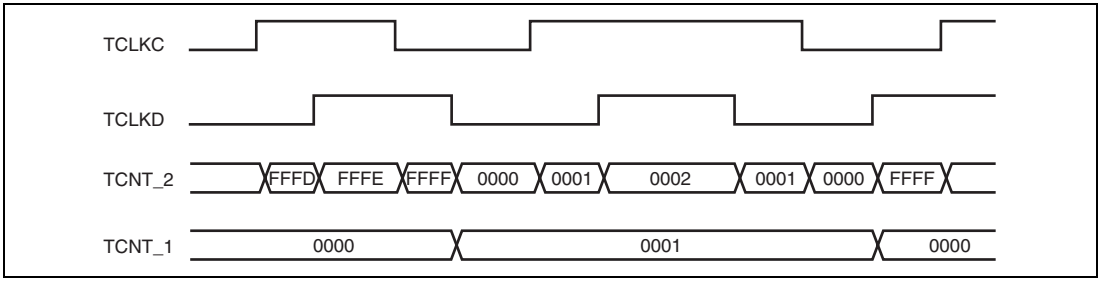


Figure 10.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 10.22 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.

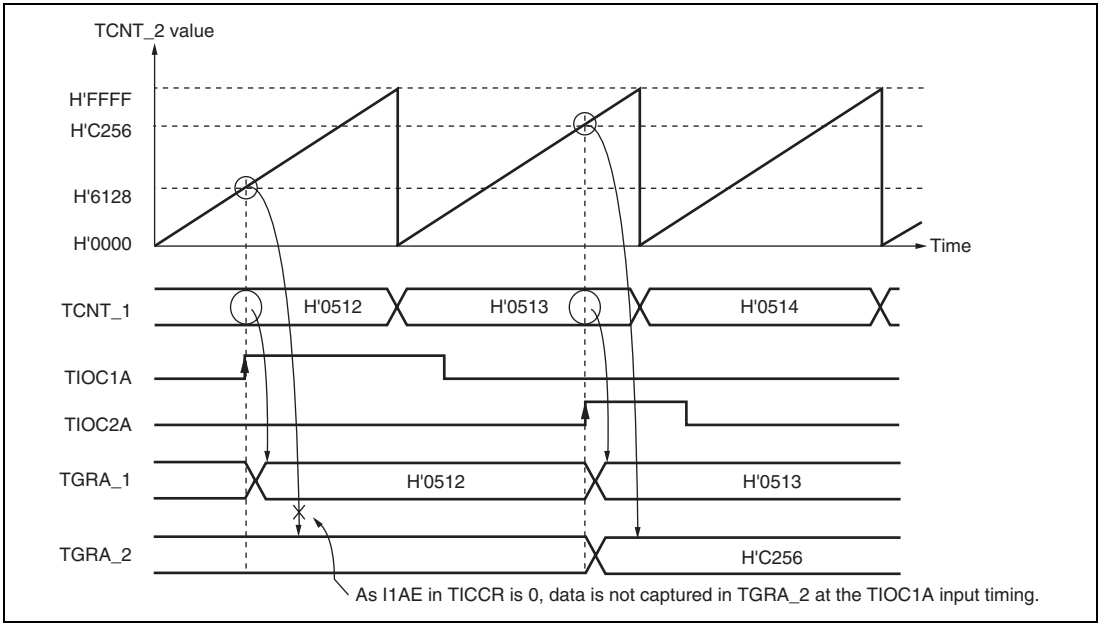


Figure 10.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 10.23 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.

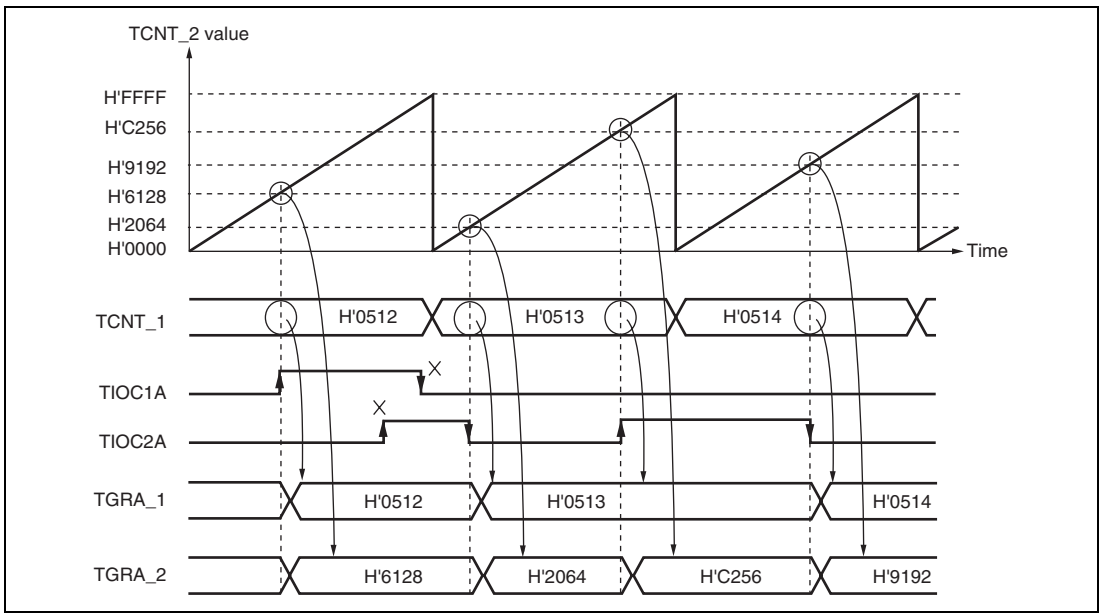


Figure 10.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 10.24 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.

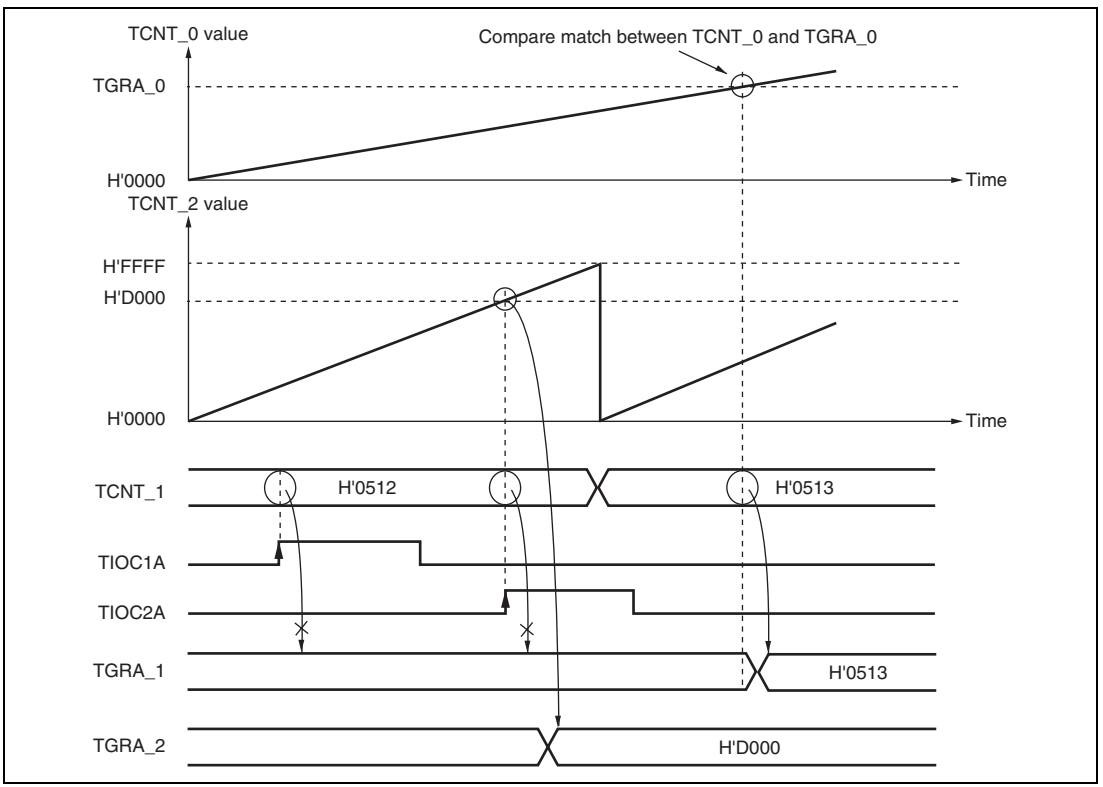


Figure 10.24 Cascaded Operation Example (d)

10.4.5 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 10.46.

Table 10.46 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 10.25 shows an example of the PWM mode setting procedure.

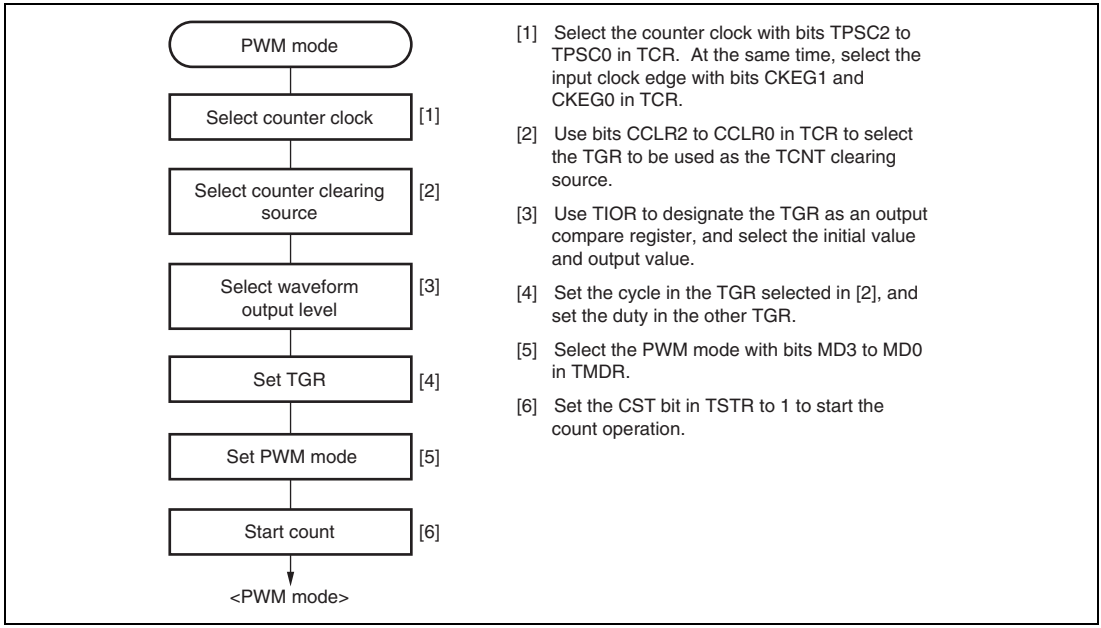


Figure 10.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 10.26 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.

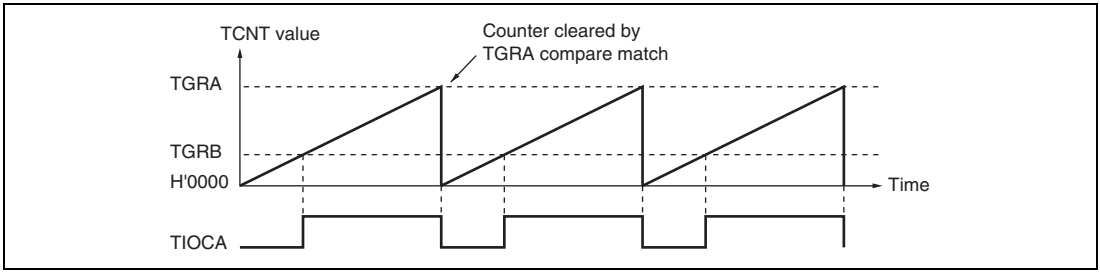


Figure 10.26 Example of PWM Mode Operation (1)

Figure 10.27 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.

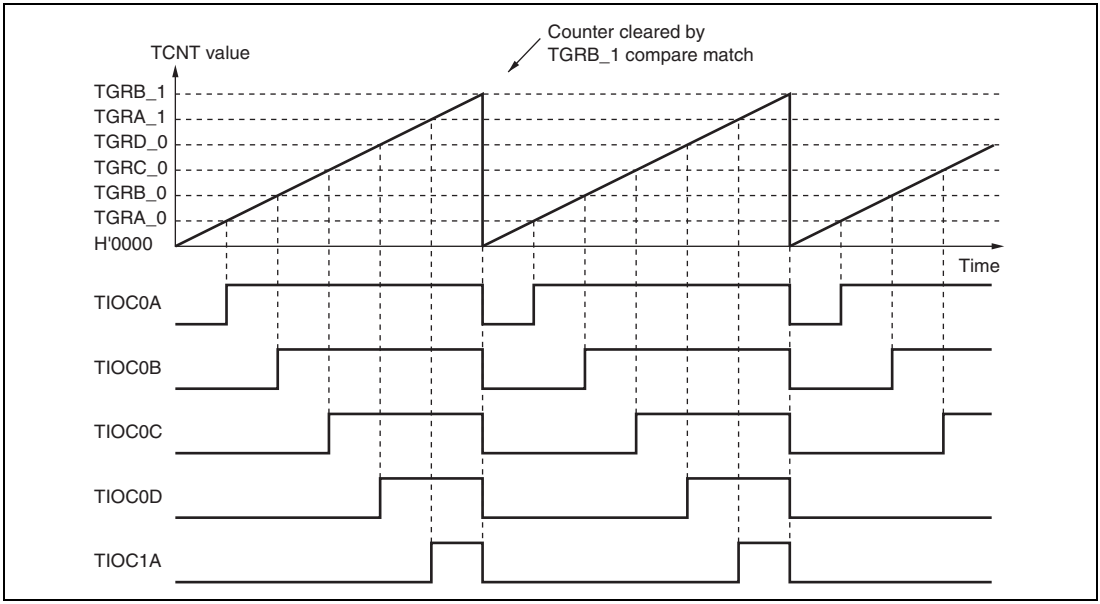


Figure 10.27 Example of PWM Mode Operation (2)

Figure 10.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

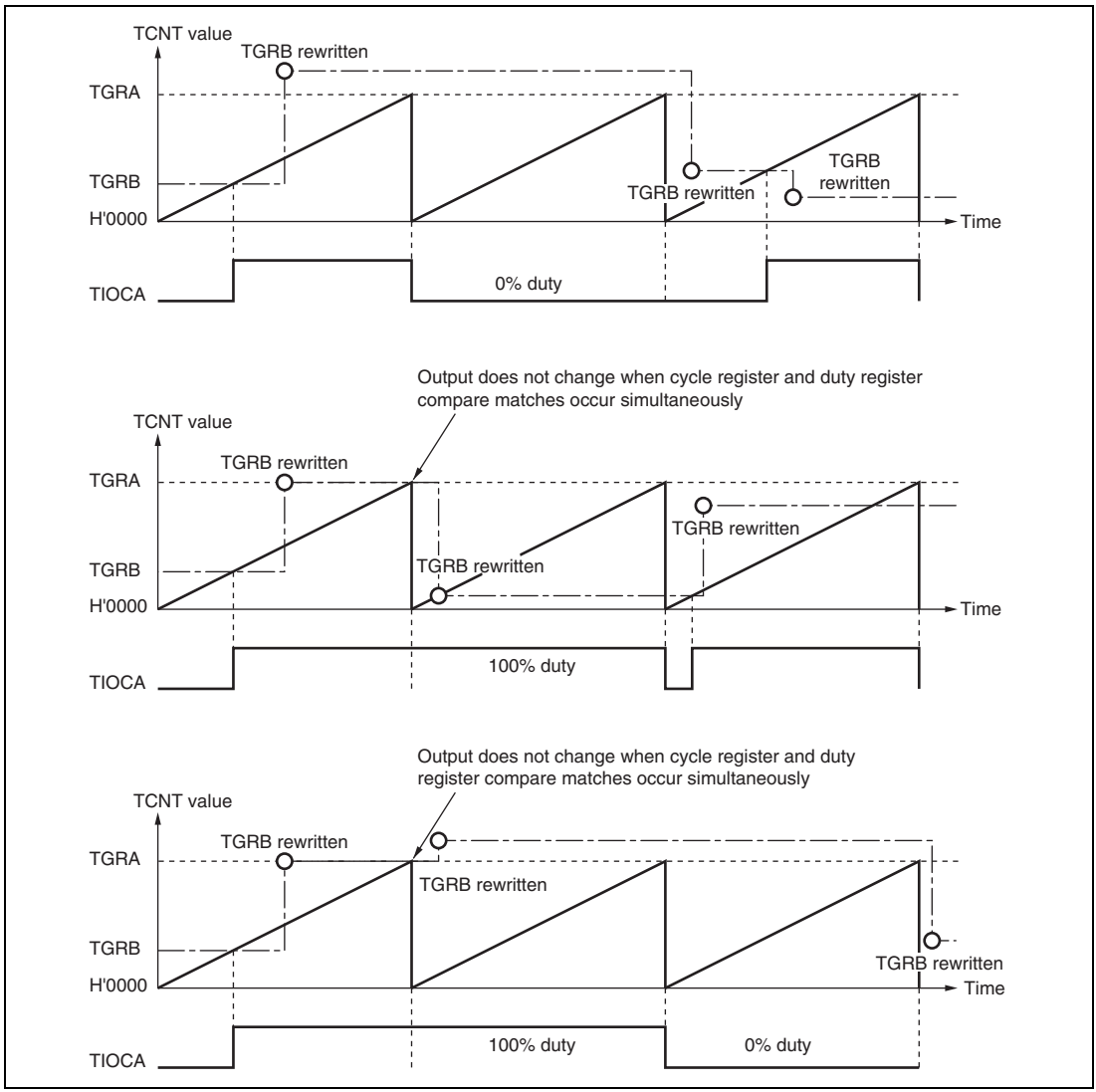


Figure 10.28 Example of PWM Mode Operation (3)

10.4.6 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 10.47 shows the correspondence between external clock pins and channels.

Table 10.47 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 10.29 shows an example of the phase counting mode setting procedure.

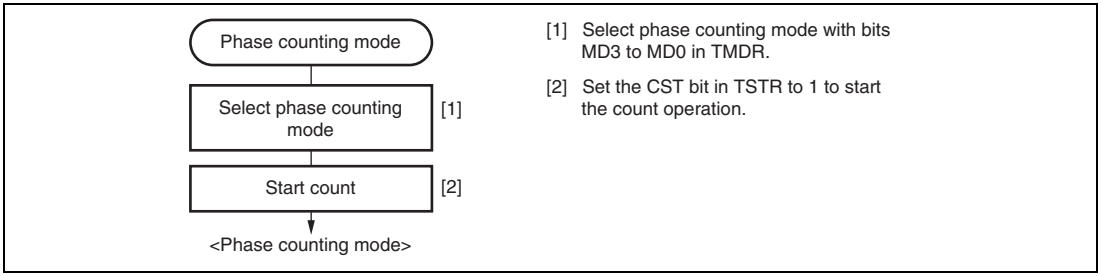


Figure 10.29 Example of Phase Counting Mode Setting Procedure

(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 10.30 shows an example of phase counting mode 1 operation, and table 10.48 summarizes the TCNT up/down-count conditions.

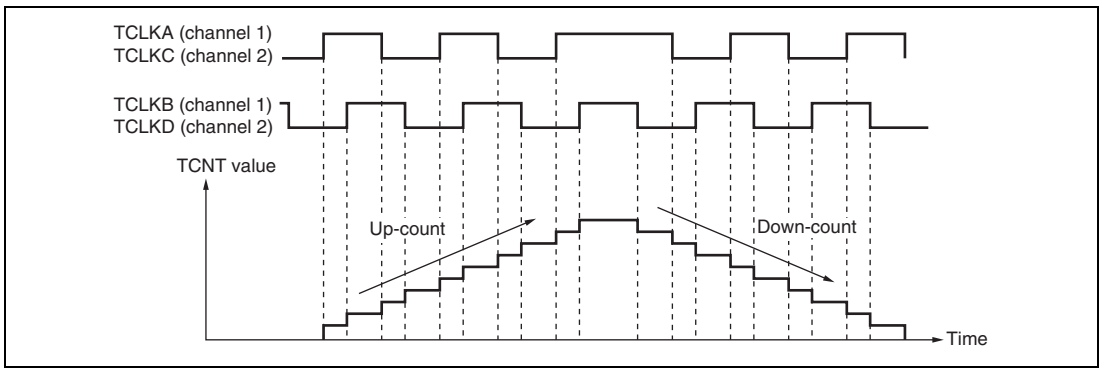


Figure 10.30 Example of Phase Counting Mode 1 Operation

Table 10.48 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 10.31 shows an example of phase counting mode 2 operation, and table 10.49 summarizes the TCNT up/down-count conditions.

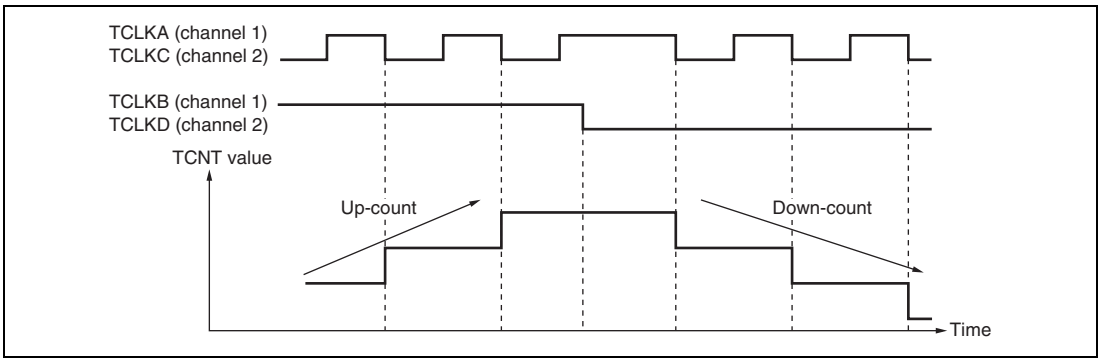


Figure 10.31 Example of Phase Counting Mode 2 Operation

Table 10.49 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 10.32 shows an example of phase counting mode 3 operation, and table 10.50 summarizes the TCNT up/down-count conditions.

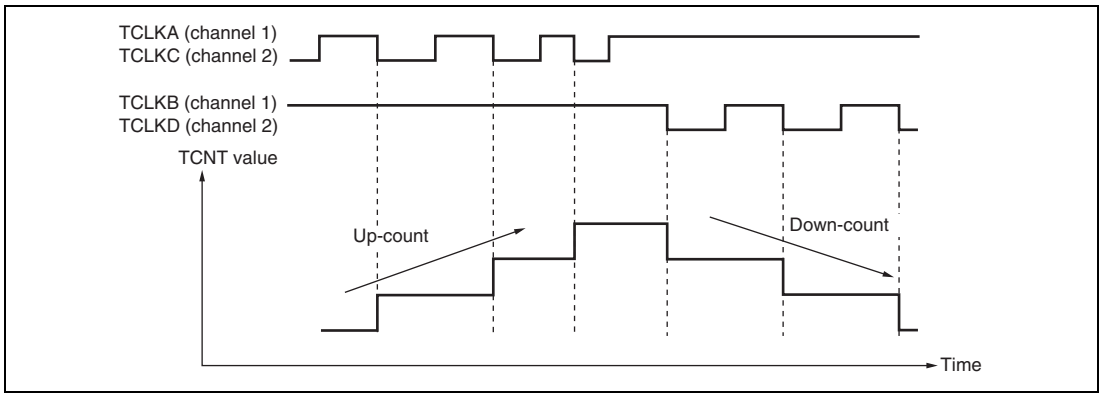


Figure 10.32 Example of Phase Counting Mode 3 Operation

Table 10.50 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 10.33 shows an example of phase counting mode 4 operation, and table 10.51 summarizes the TCNT up/down-count conditions.

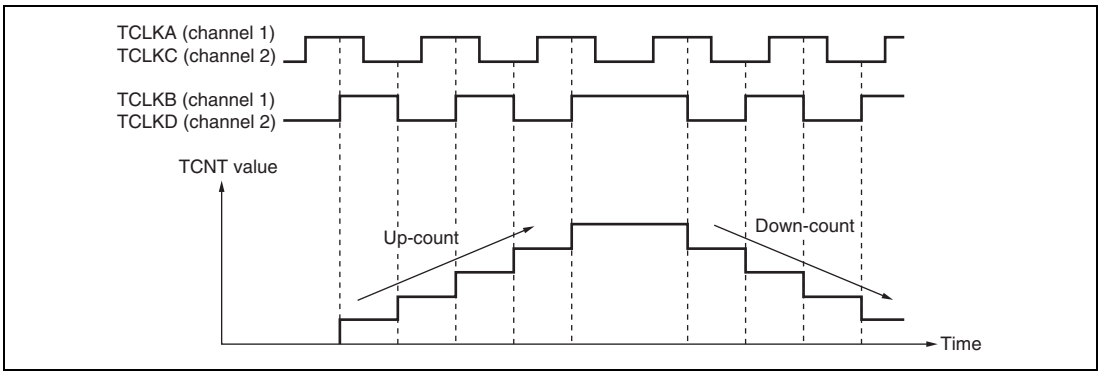


Figure 10.33 Example of Phase Counting Mode 4 Operation

Table 10.51 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 10.34 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.

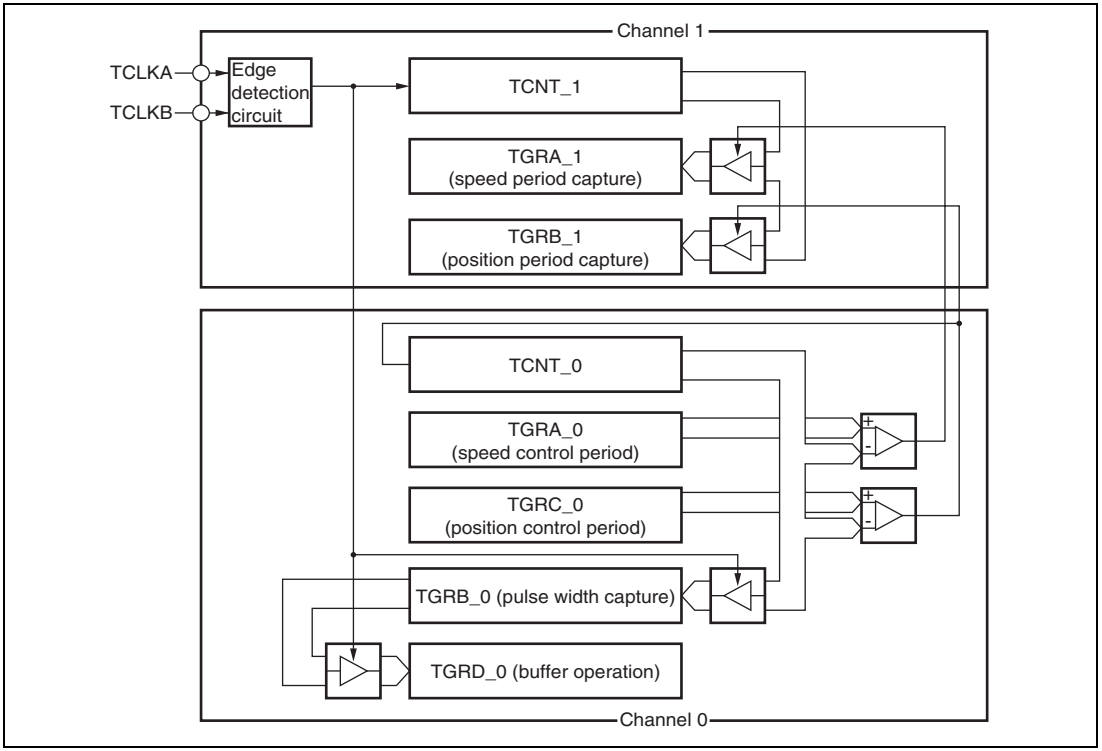


Figure 10.34 Phase Counting Mode Application Example

10.4.7 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 10.52 shows the PWM output pins used. Table 10.53 shows the settings of the registers.

Table 10.52 Output Pins for Reset-Synchronized PWM Mode

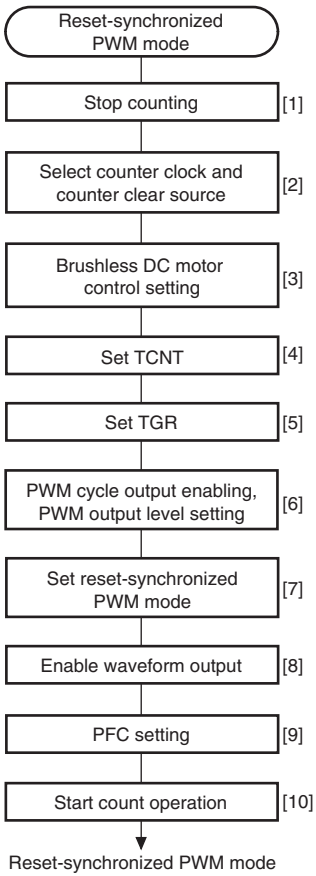
Channel	Output Pin	Description
3	TIOC3B	PWM output pin 1
	TIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
4	TIOC4A	PWM output pin 2
	TIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	TIOC4B	PWM output pin 3
	TIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 10.53 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 10.35 shows an example of procedure for selecting reset-synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in the TSTR to 0 to halt the counting of TCNT. The reset-synchronized PWM mode must be set up while TCNT_3 and TCNT_4 are halted.

[2] Set bits TPSC2-TPSC0 and CKEG1 and CKEG0 in the TCR_3 to select the counter clock and clock edge for channel 3. Set bits CCLR2-CCLR0 in the TCR_3 to select TGRA compare-match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCR) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Reset TCNT_3 and TCNT_4 to H'0000.

[5] TGRA_3 is the period register. Set the waveform period value in TGRA_3. Set the transition timing of the PWM output waveforms in TGRB_3, TGRA_4, and TGRB_4. Set times within the compare-match range of TCNT_3.
$X \leq$ TGRA_3 (X: set value).

[6] Select enabling/disabling of toggle output synchronized with the PMW cycle using bit PSYE in the timer output control register (TOCR), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBR as a buffer for TOCR_2, see figure 10.3.

[7] Set bits MD3-MD0 in TMDR_3 to B'1000 to select the reset-synchronized PWM mode. Do not set to TMDR_4.

[8] Set the enabling/disabling of the PWM waveform output pin in TOER.

[9] Set the port control register and the port I/O register.

[10] Set the CST3 bit in the TSTR to 1 to start the count operation.

Note: The output waveform starts to toggle operation at the point of TCNT_3 = TGRA_3 = X by setting X = TGRA, i.e., cycle = duty.

Figure 10.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Reset-Synchronized PWM Mode Operation

Figure 10.36 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.

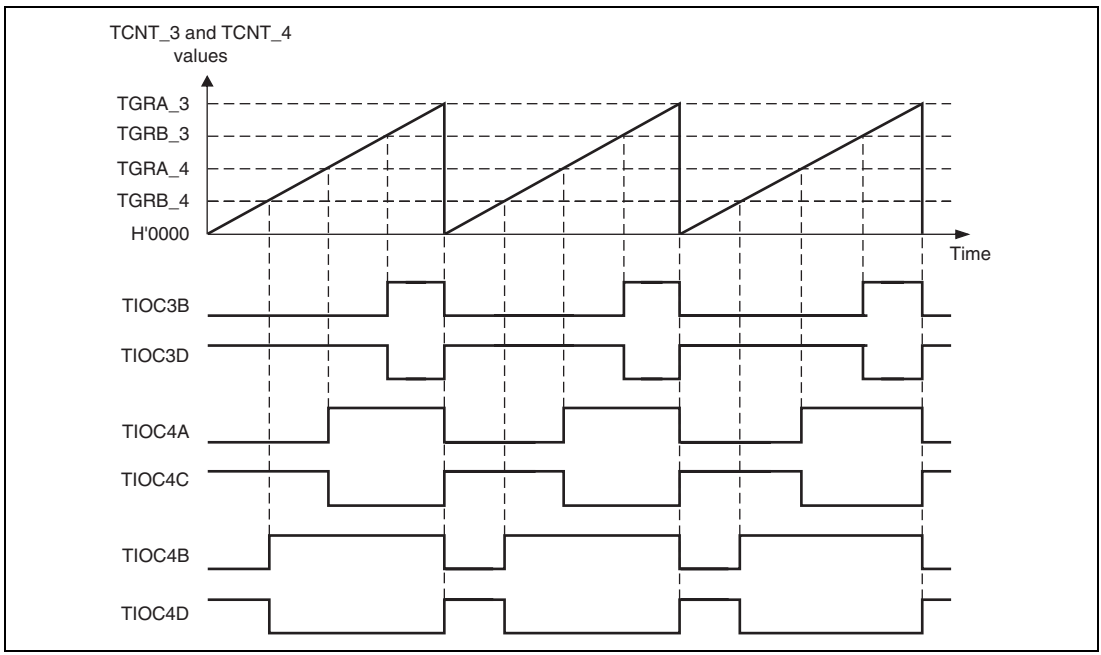


Figure 10.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

10.4.8 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 10.54 shows the PWM output pins used. Table 10.55 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 10.54 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 10.55 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).

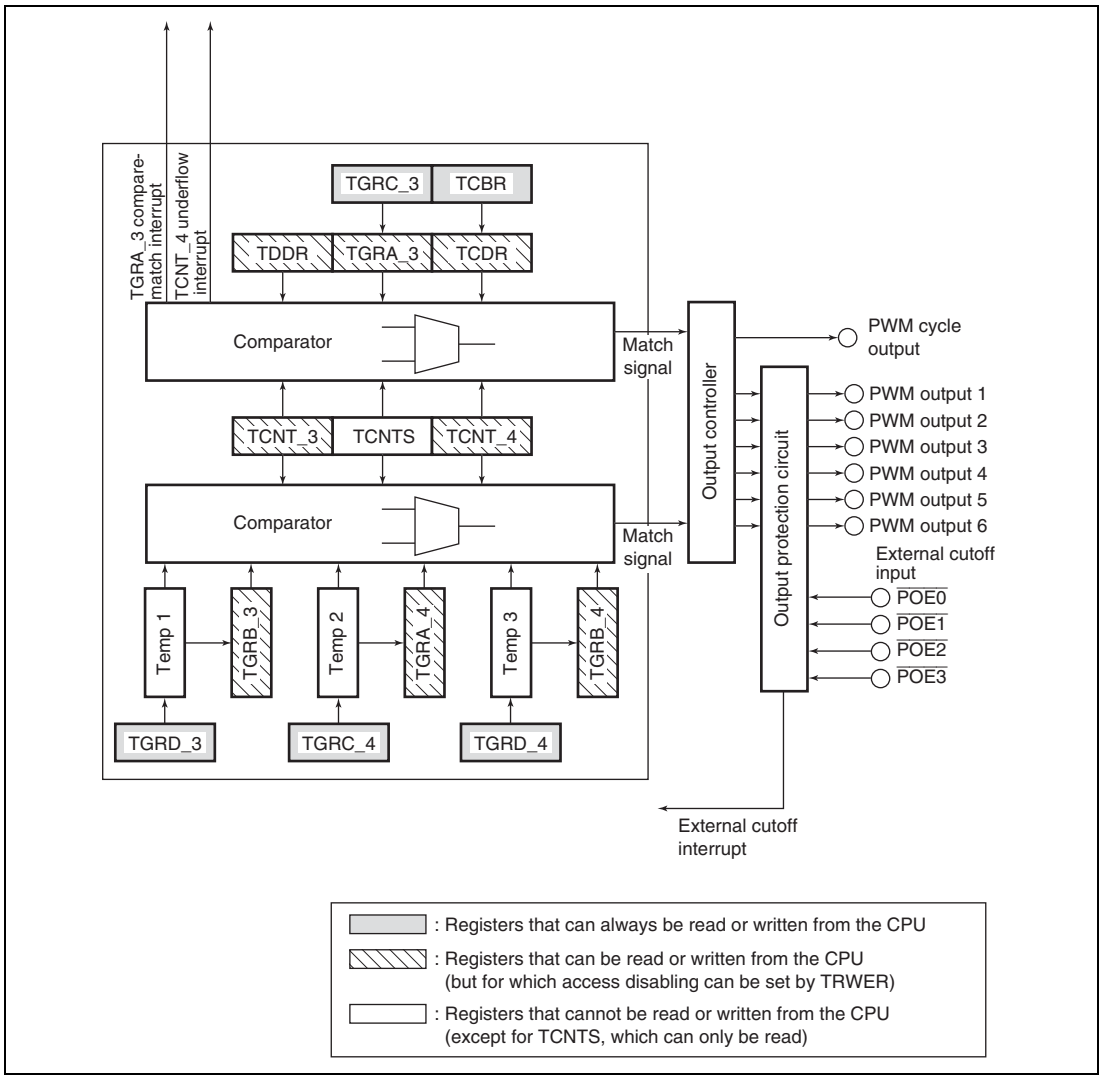
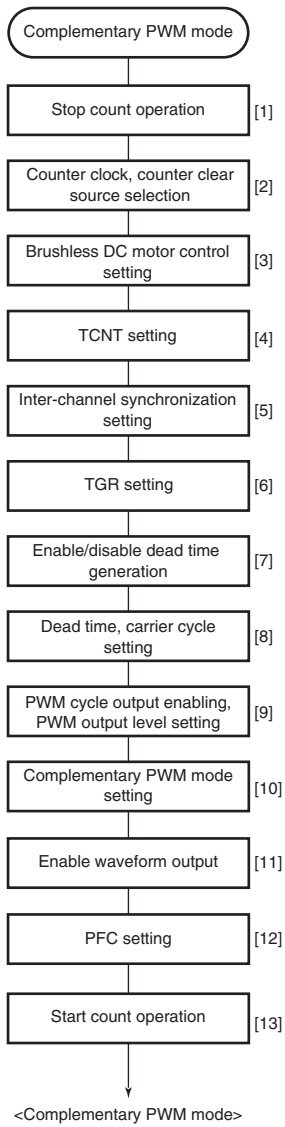


Figure 10.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 10.38.



[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 10.38 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 10.39 illustrates counter operation in complementary PWM mode, and figure 10.40 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.

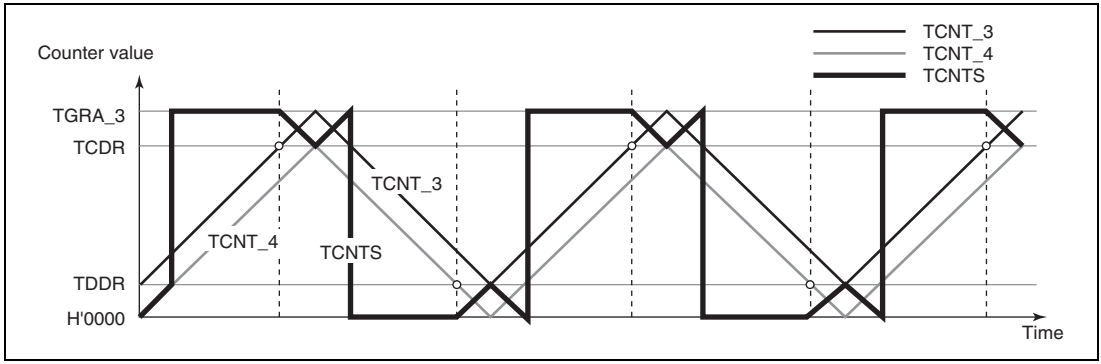


Figure 10.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 10.40 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 10.40 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 10.40) 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.

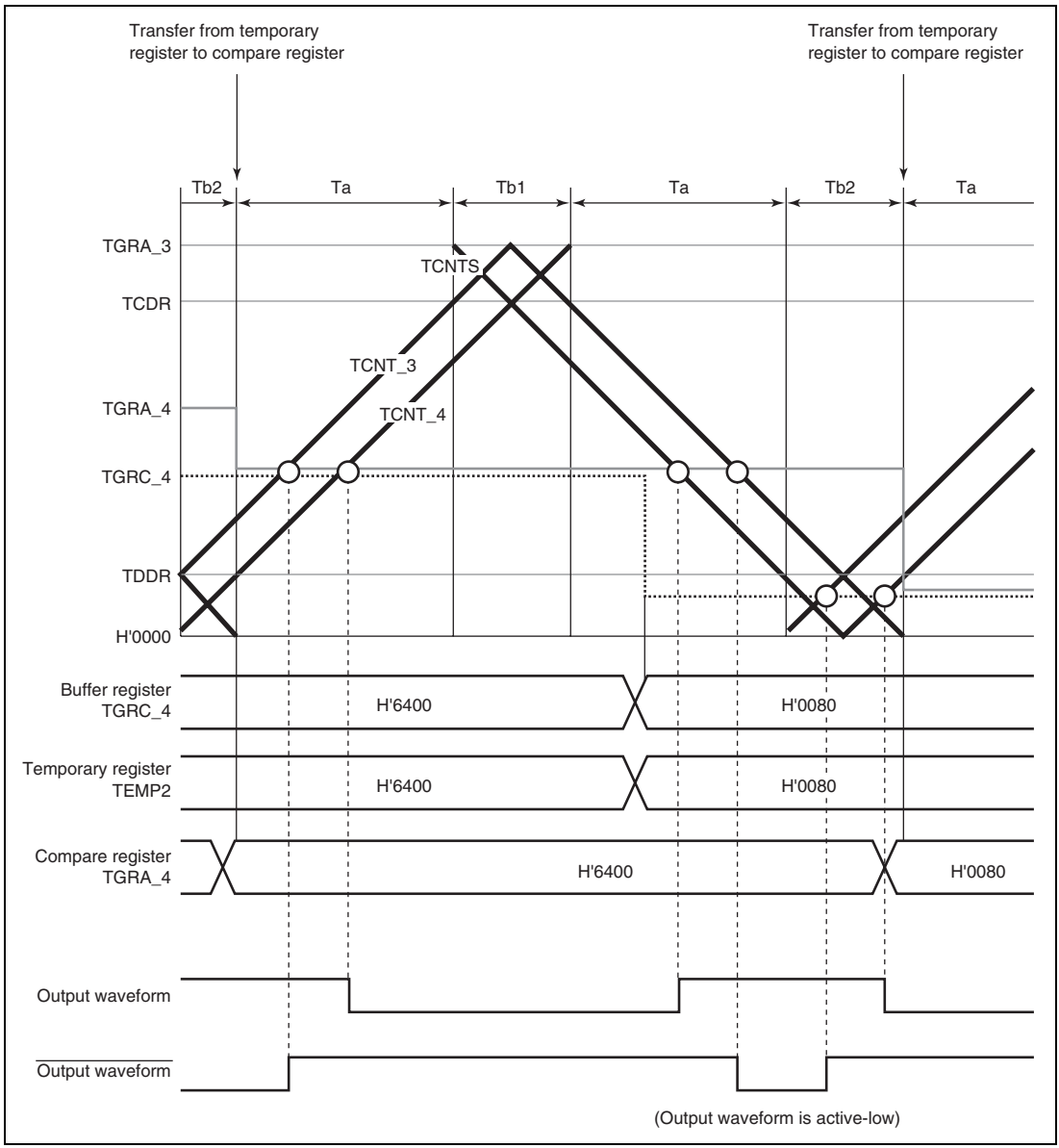


Figure 10.40 Example of Complementary PWM Mode Operation

(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 10.56 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 10.41 shows an example of operation without dead time.

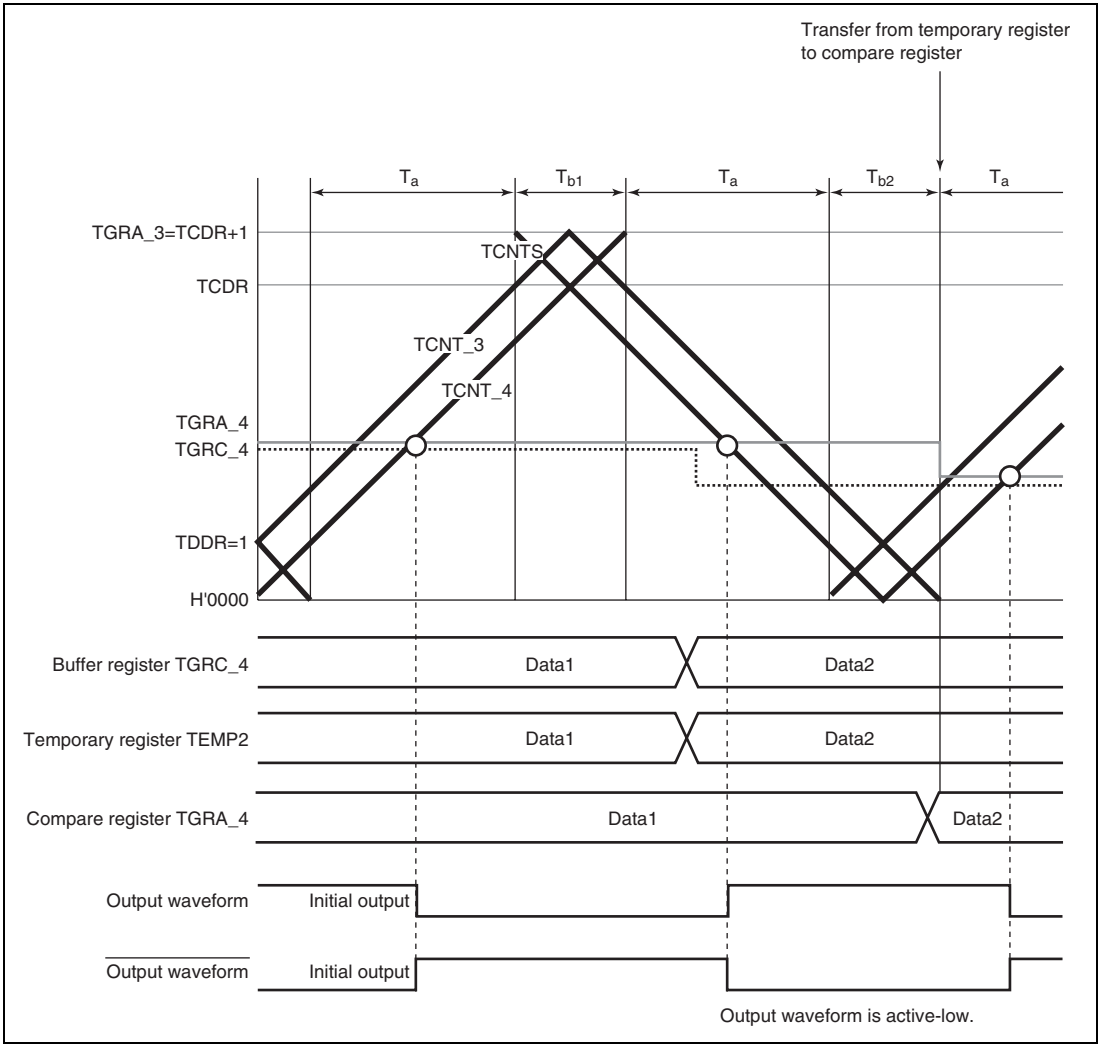


Figure 10.41 Example of Operation without Dead Time

(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 10.42 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.

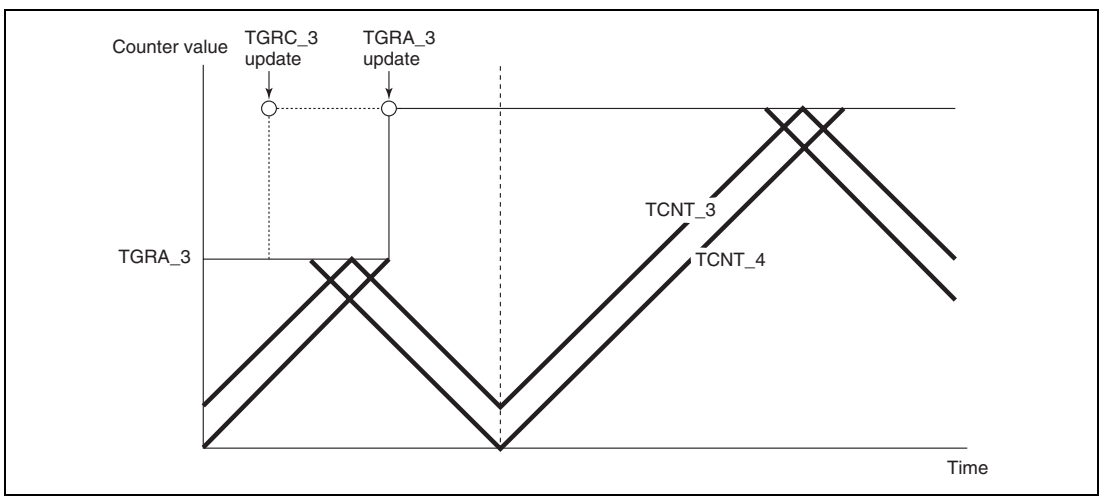


Figure 10.42 Example of PWM Cycle Updating

(h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five PWM duty and carrier cycle registers that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. When subcounter TCNTS is not counting, if buffer register data is updated, the temporary register value is also rewritten. Transfer is not performed from buffer registers to temporary registers when TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD3 to MD0 in the timer mode register (TMDR). Figure 10.43 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.

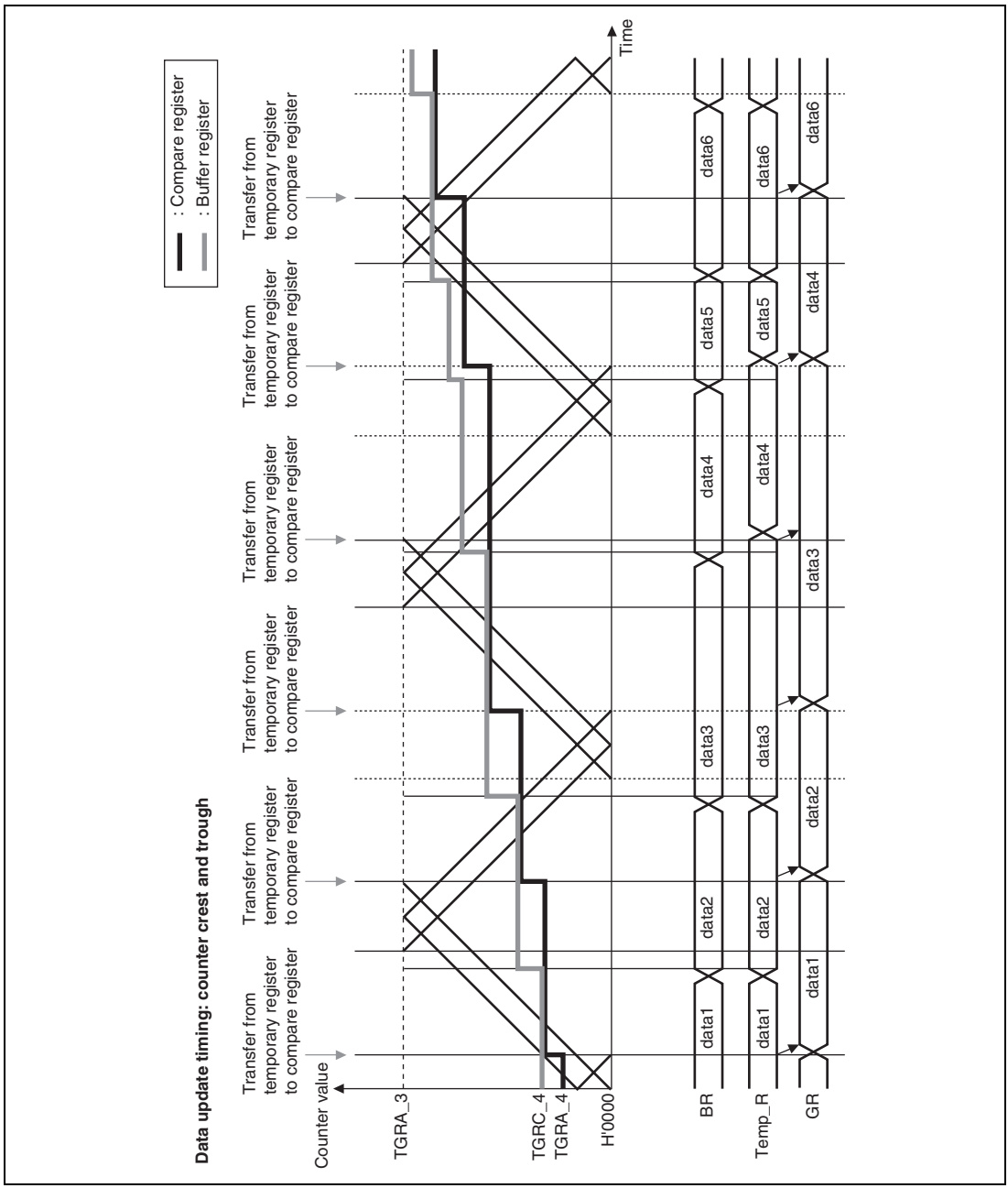


Figure 10.43 Example of Data Update in Complementary PWM Mode

(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 10.44 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 10.45.

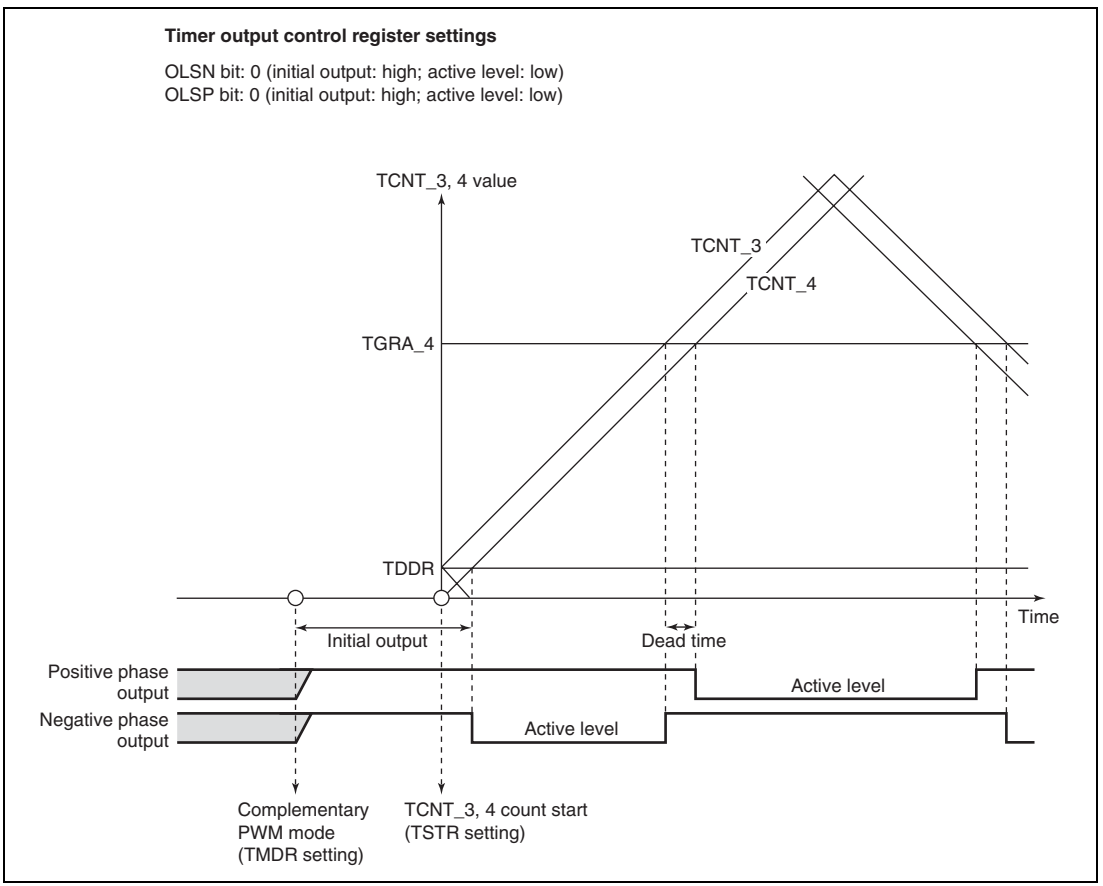


Figure 10.44 Example of Initial Output in Complementary PWM Mode (1)

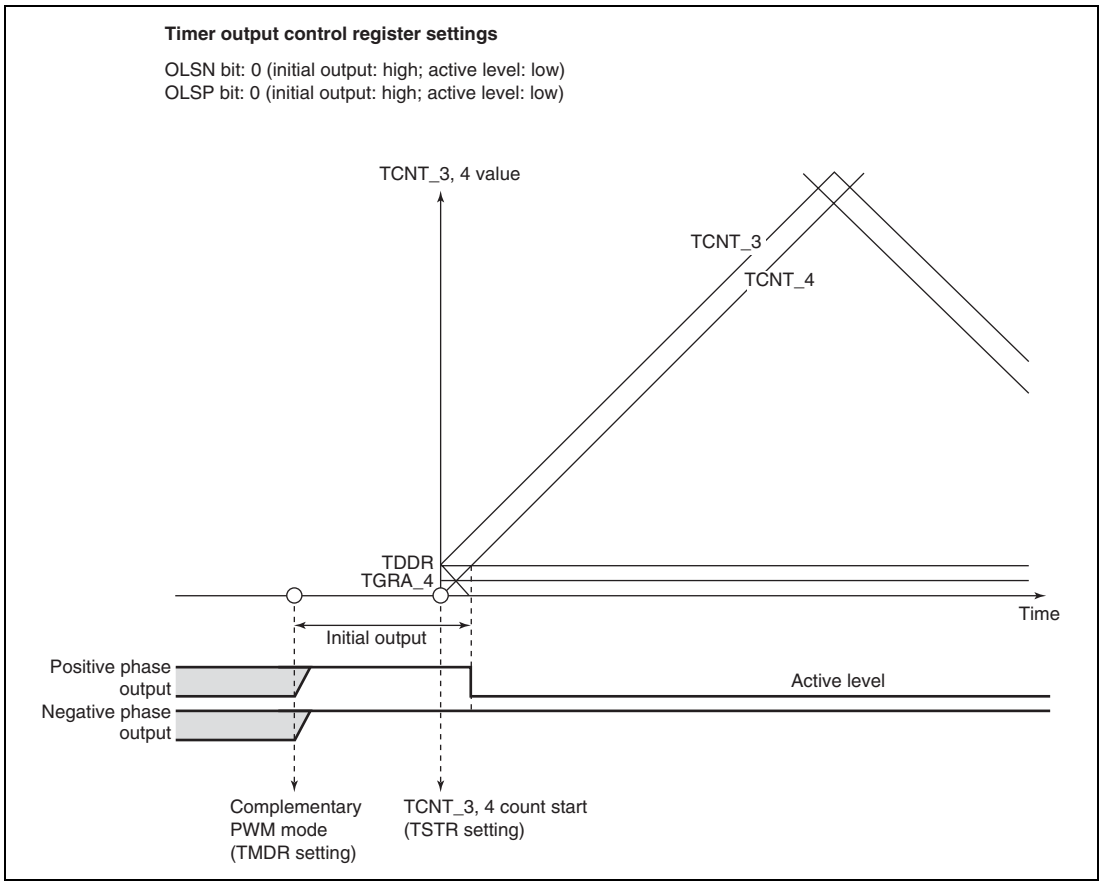


Figure 10.45 Example of Initial Output in Complementary PWM Mode (2)

(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 10.46 to 10.48 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 10.46.

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 10.47, 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 10.48, 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.

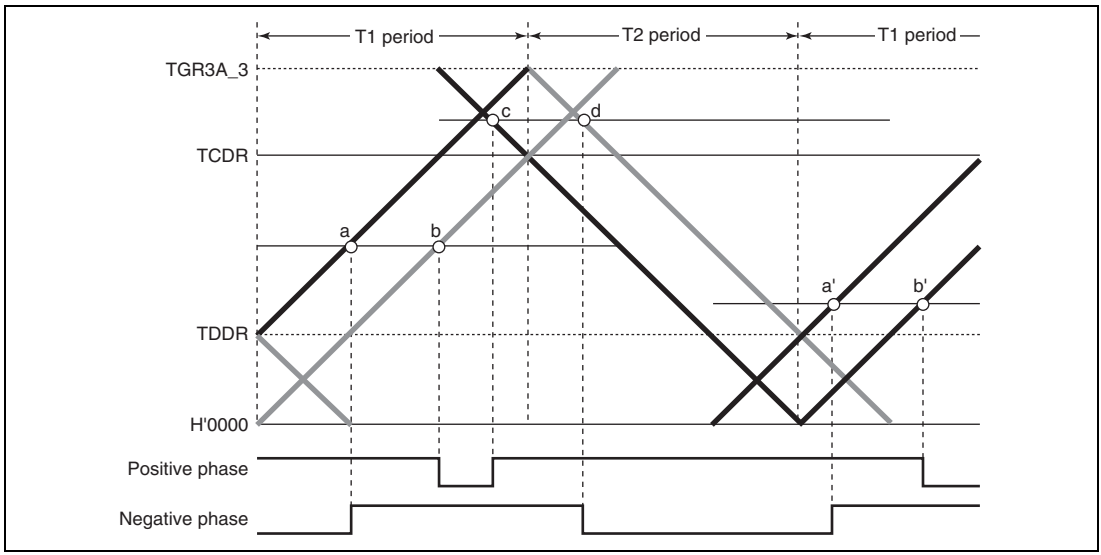


Figure 10.46 Example of Complementary PWM Mode Waveform Output (1)

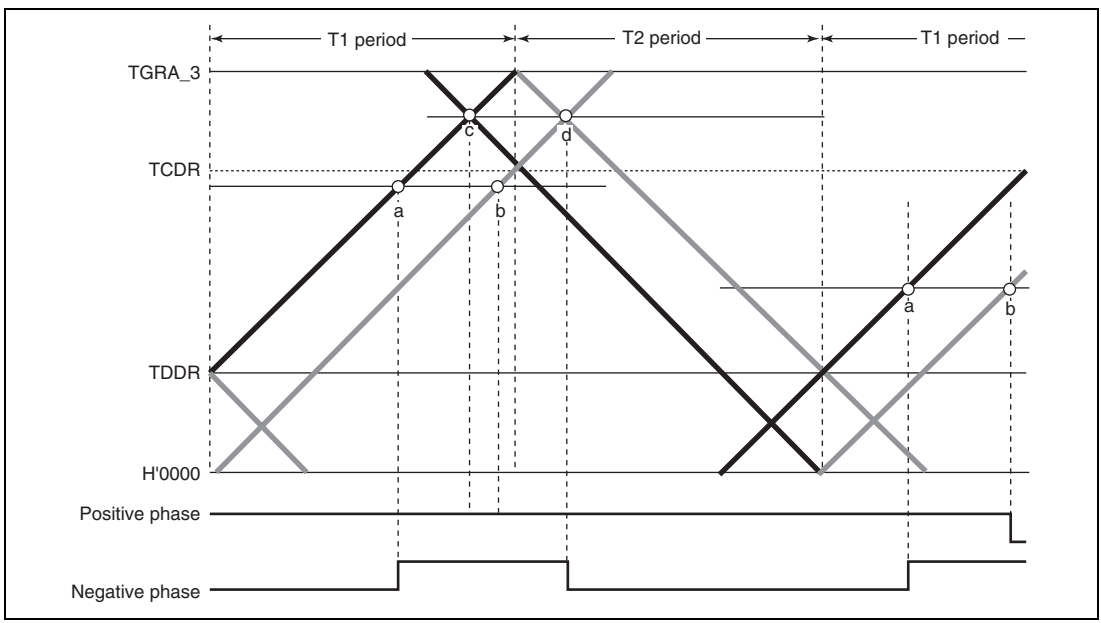


Figure 10.47 Example of Complementary PWM Mode Waveform Output (2)

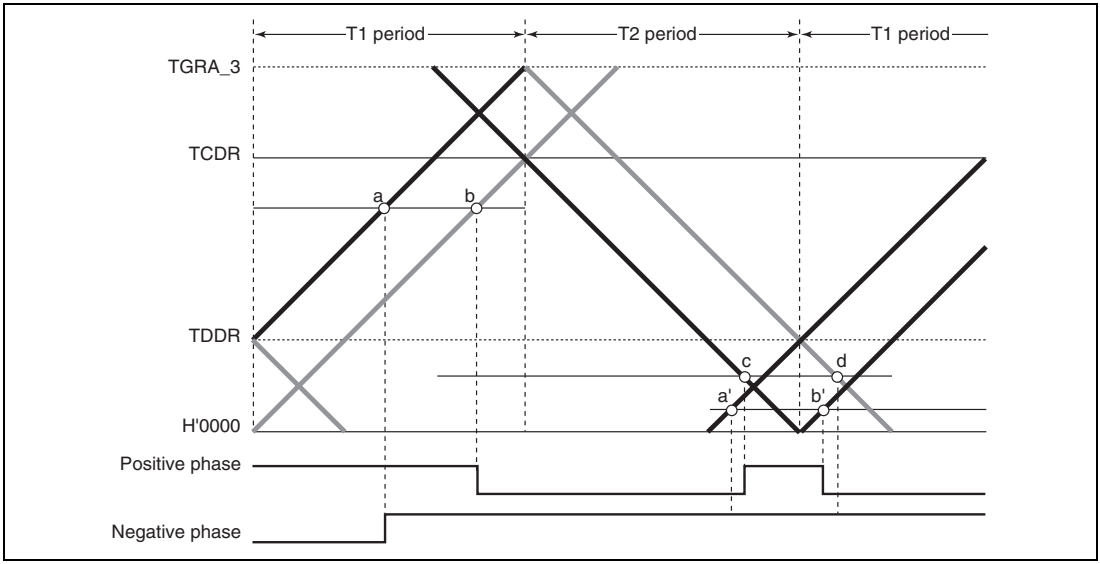


Figure 10.48 Example of Complementary PWM Mode Waveform Output (3)

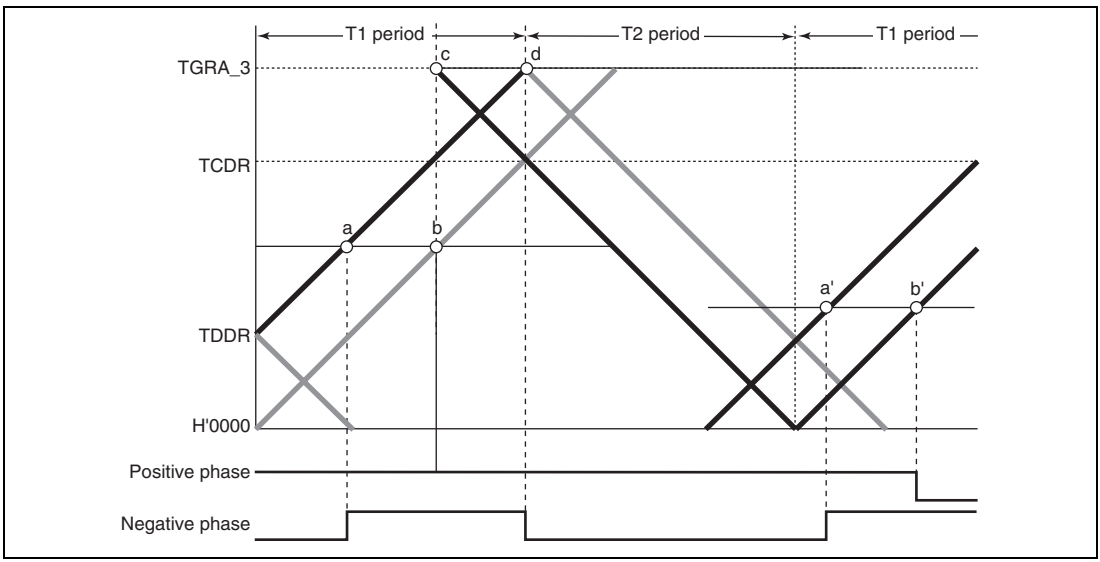


Figure 10.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

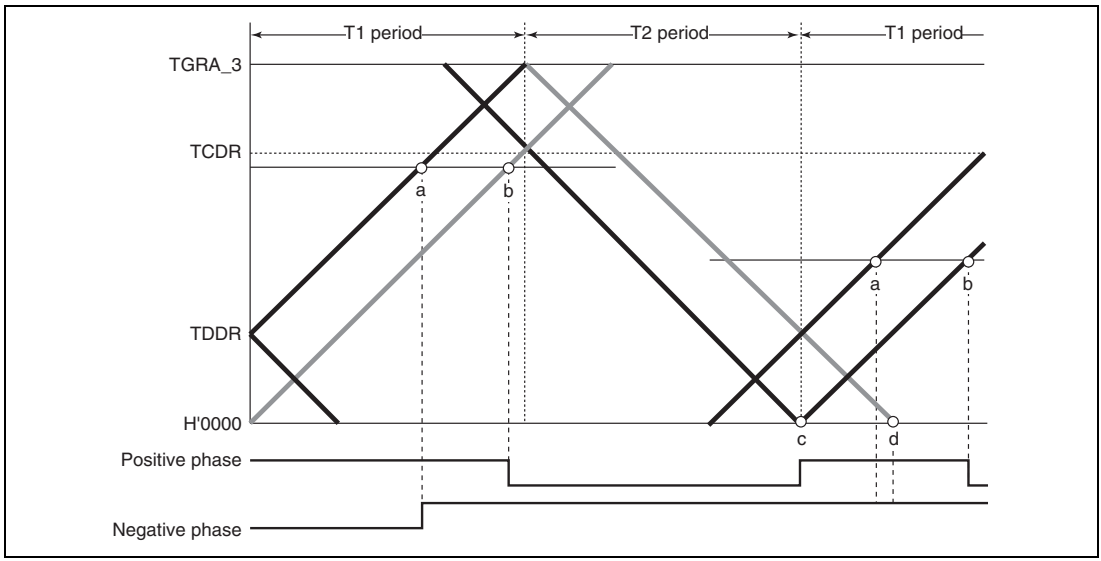


Figure 10.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

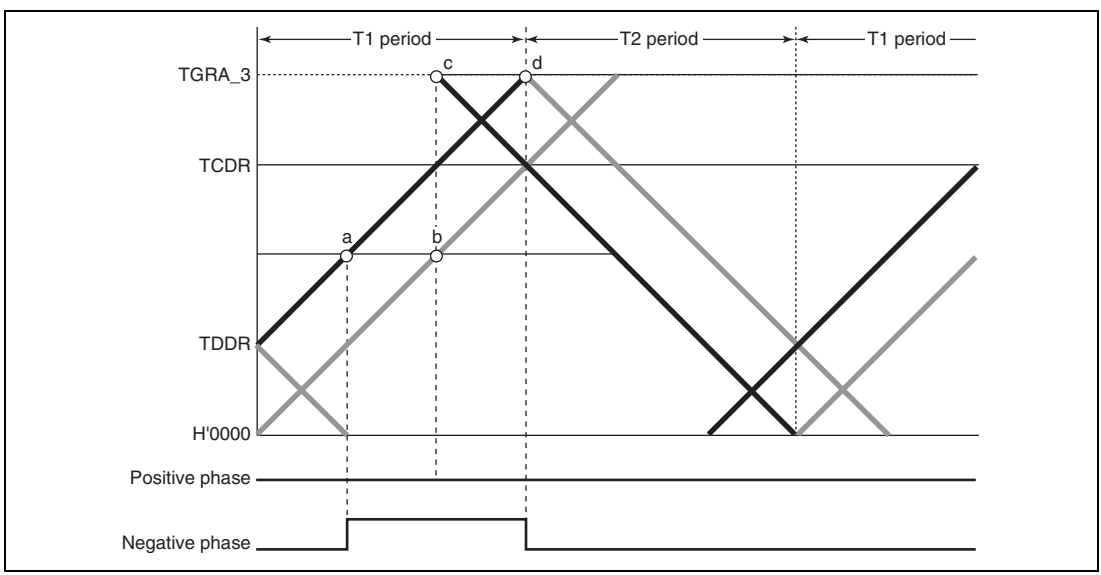


Figure 10.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

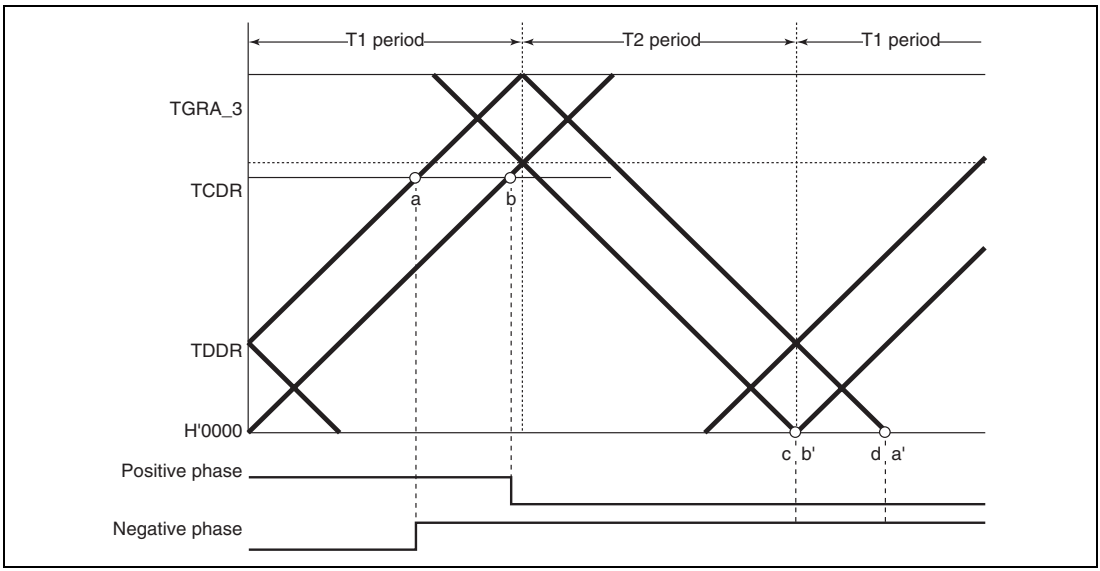


Figure 10.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

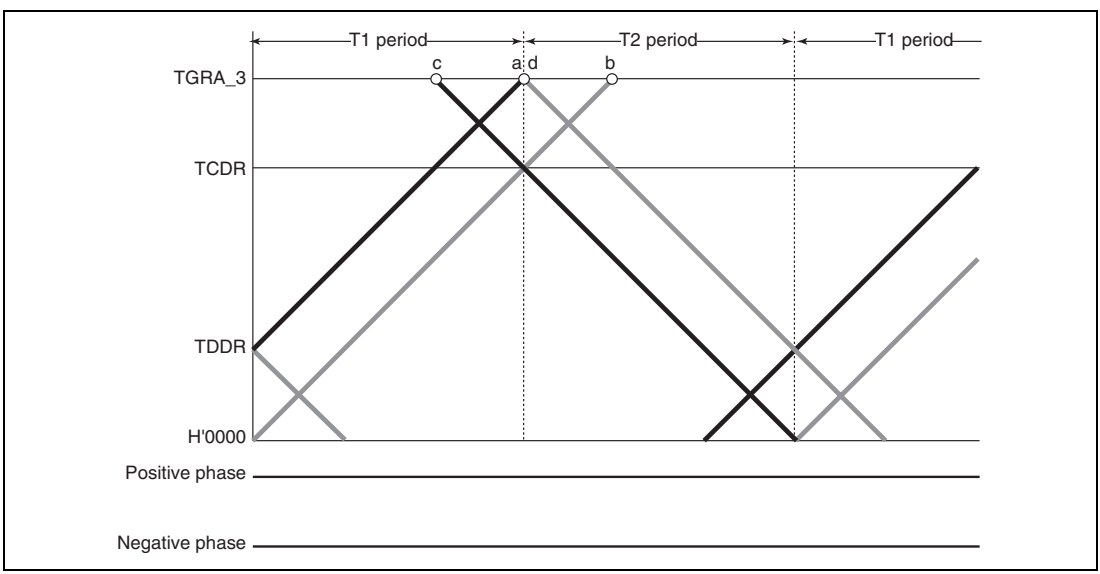


Figure 10.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(k) Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 10.49 to 10.53 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.

(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 10.54.

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.

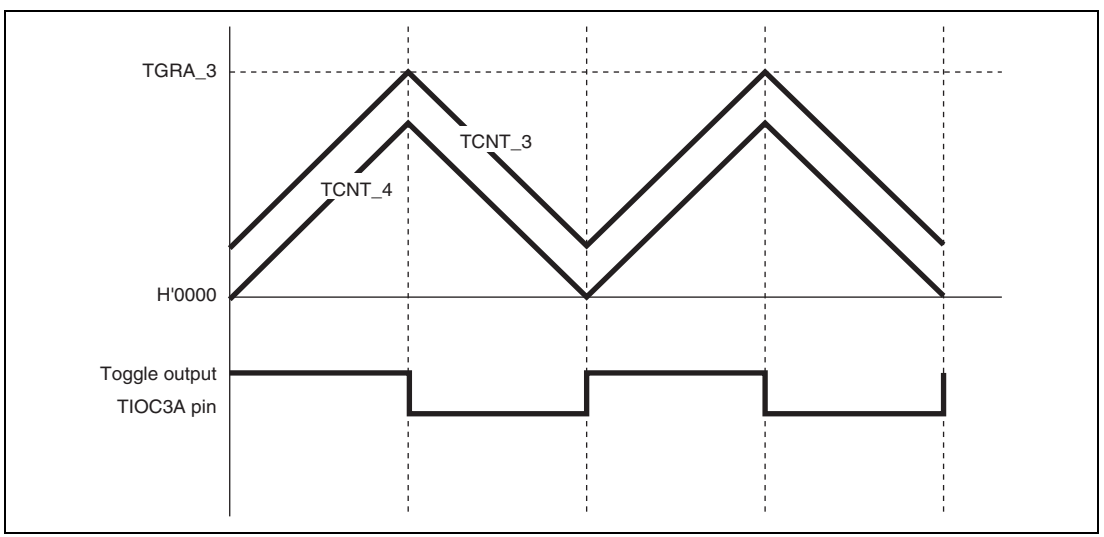


Figure 10.54 Example of Toggle Output Waveform Synchronized with PWM Output

(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 10.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

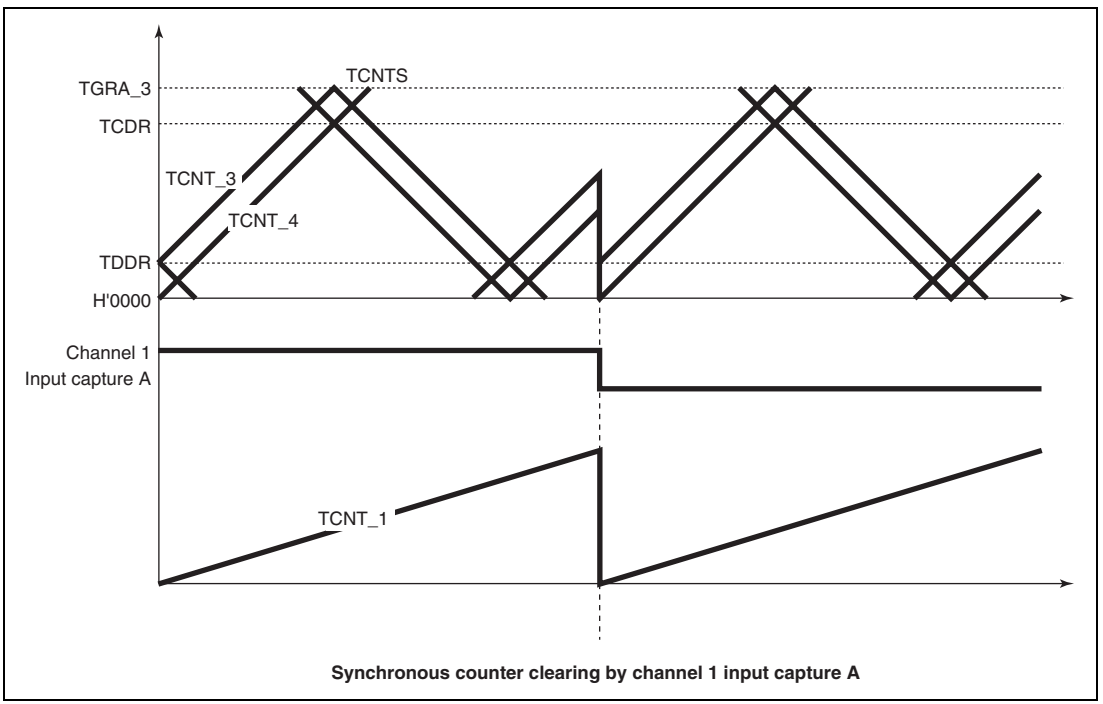


Figure 10.55 Counter Clearing Synchronized with Another Channel

(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 10.56. 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 10.56) 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.

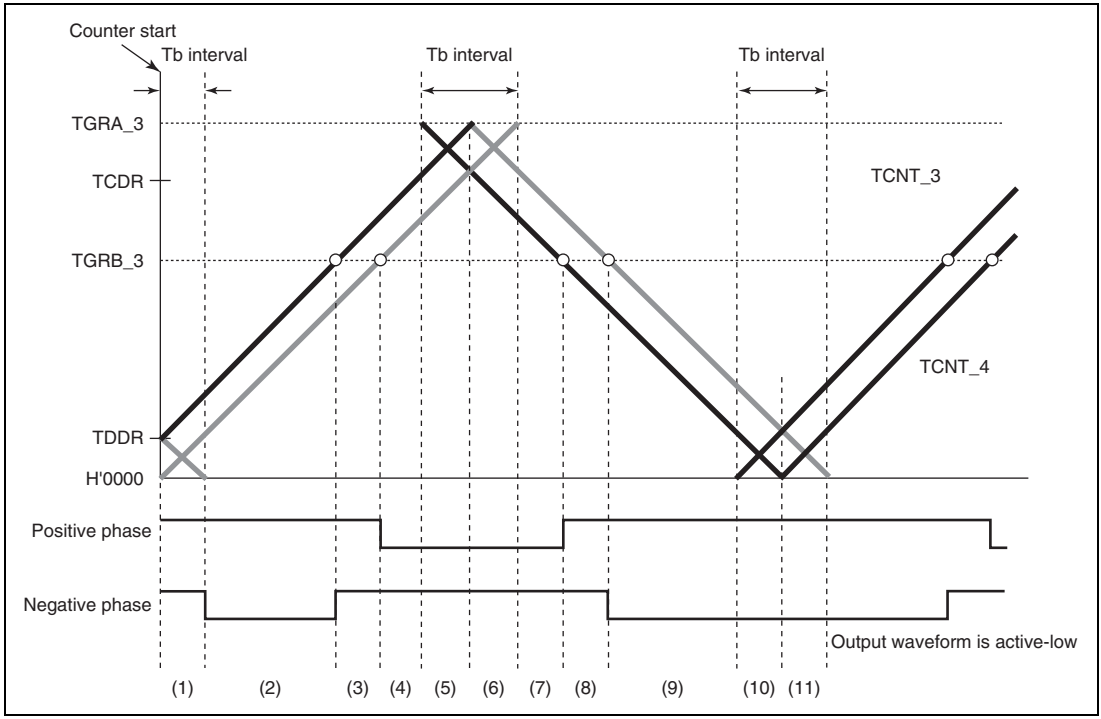


Figure 10.56 Timing for Synchronous Counter Clearing

- 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 10.57.

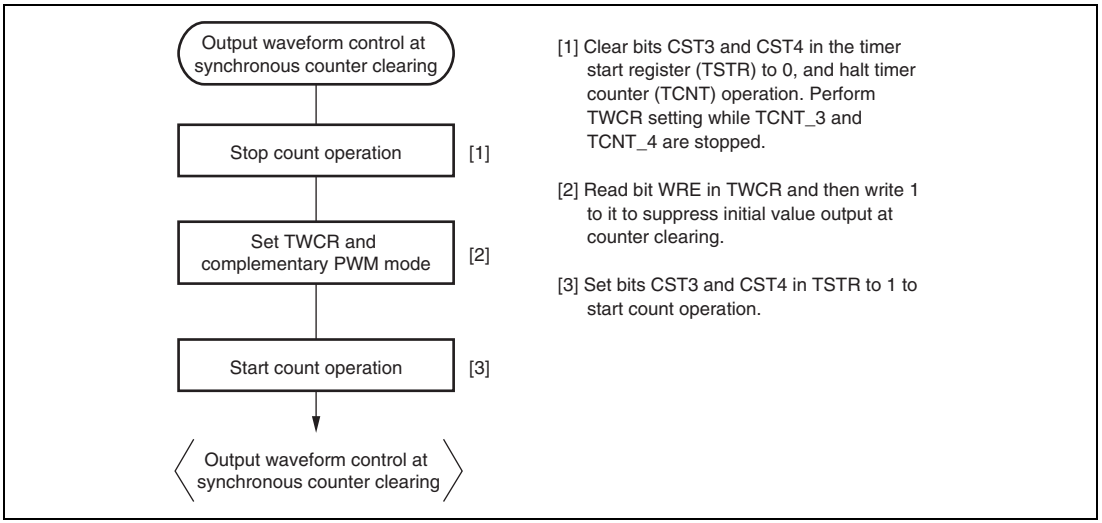


Figure 10.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 10.58 to 10.61 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 10.58 to 10.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 10.56, 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.

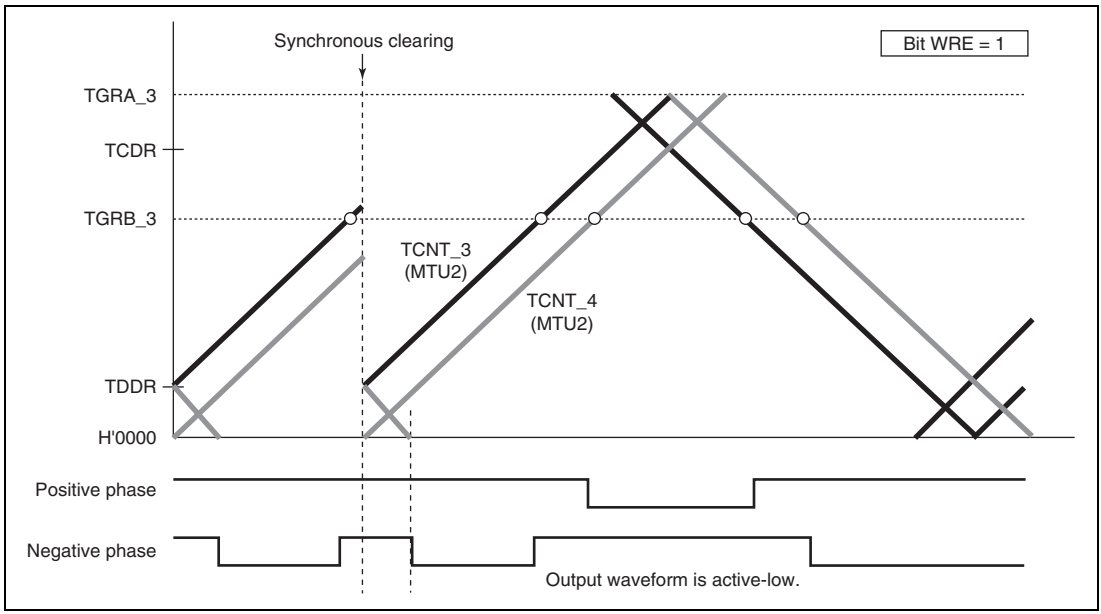


Figure 10.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)

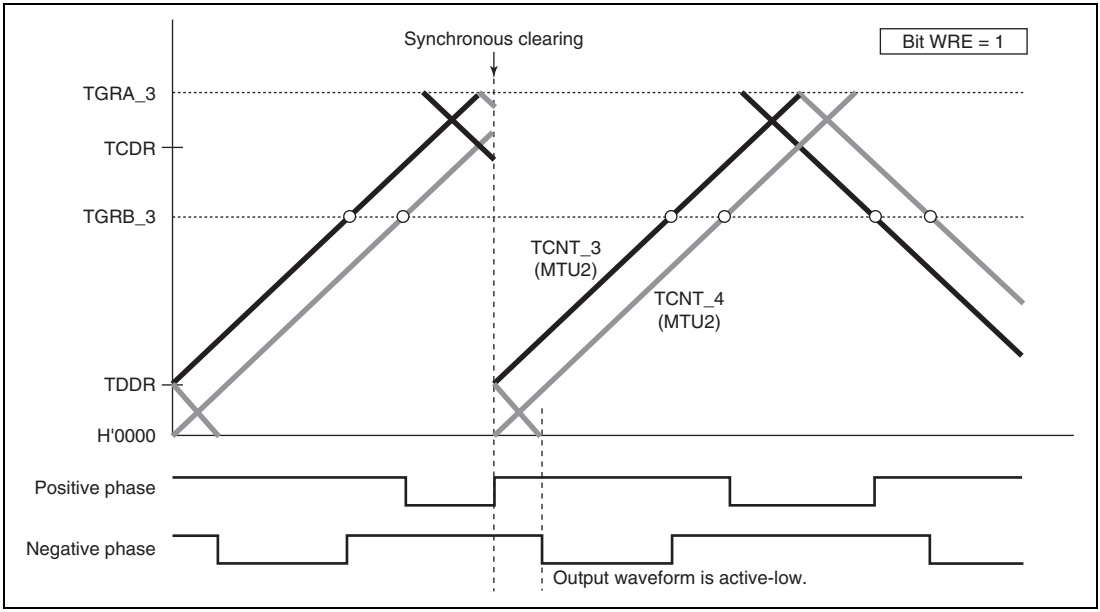


Figure 10.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE of TWCR in MTU2 is 1)

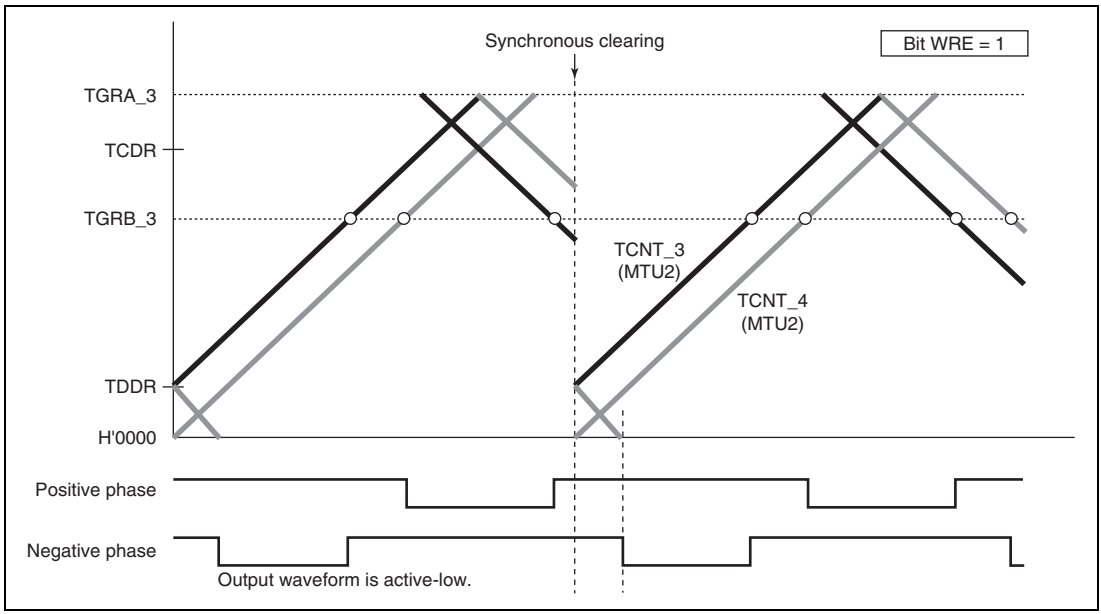


Figure 10.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 10.56; Bit WRE of TWCR is 1)

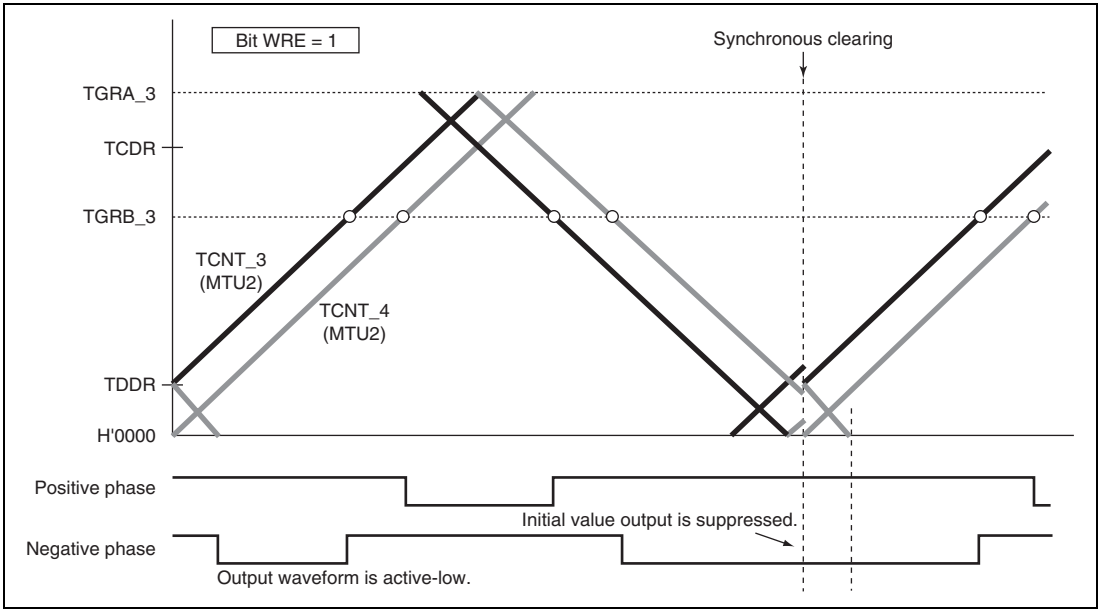


Figure 10.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE of TWCR is 1)

(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 10.62. 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 10.4.10, MTU2–MTU2S Synchronous Operation.

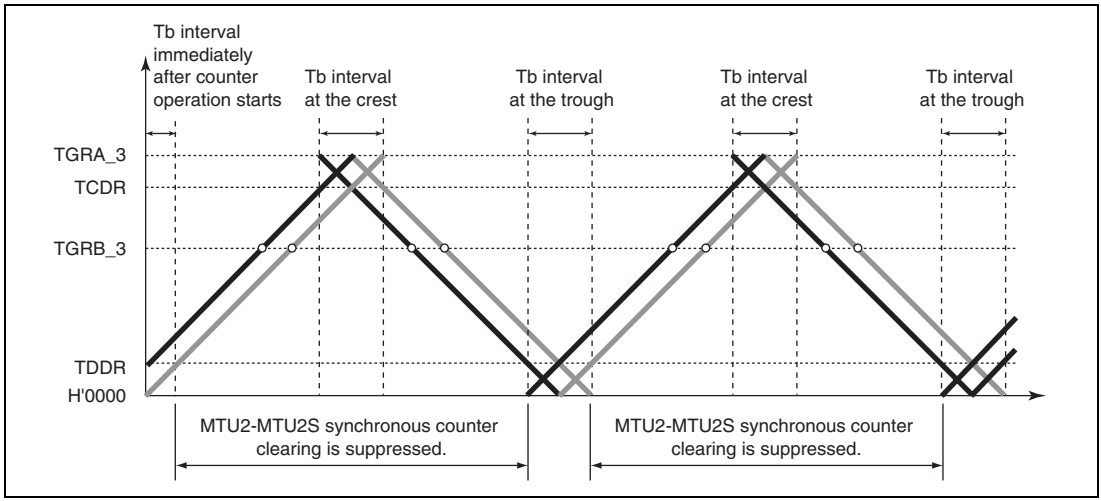


Figure 10.62 MTU2–MTU2S Synchronous Clearing-Suppressed Interval Specified by SCC Bit in TWCR

- 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 10.63.

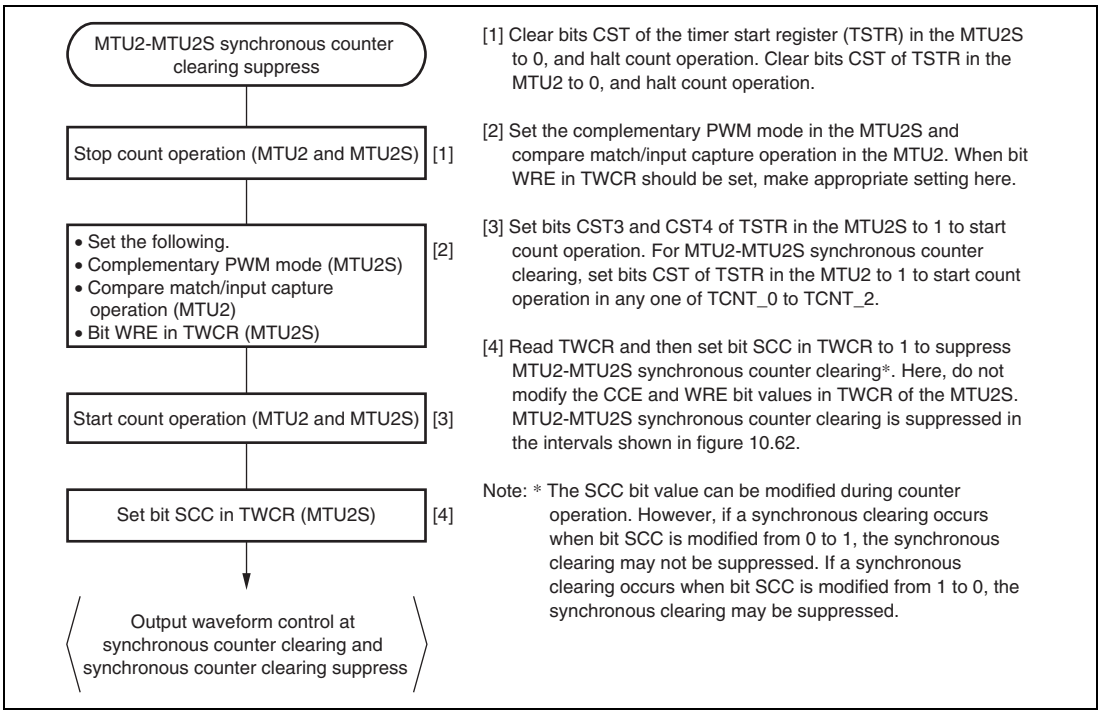


Figure 10.63 Example of Procedure for Suppressing MTU2–MTU2S Synchronous Counter Clearing

- Examples of Suppression of MTU2–MTU2S Synchronous Counter Clearing

  Figures 10.64 to 10.67 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 10.64 to 10.67, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 10.56, respectively.

  In these examples, the WRE bit in TWCR of the MTU2S is set to 1.

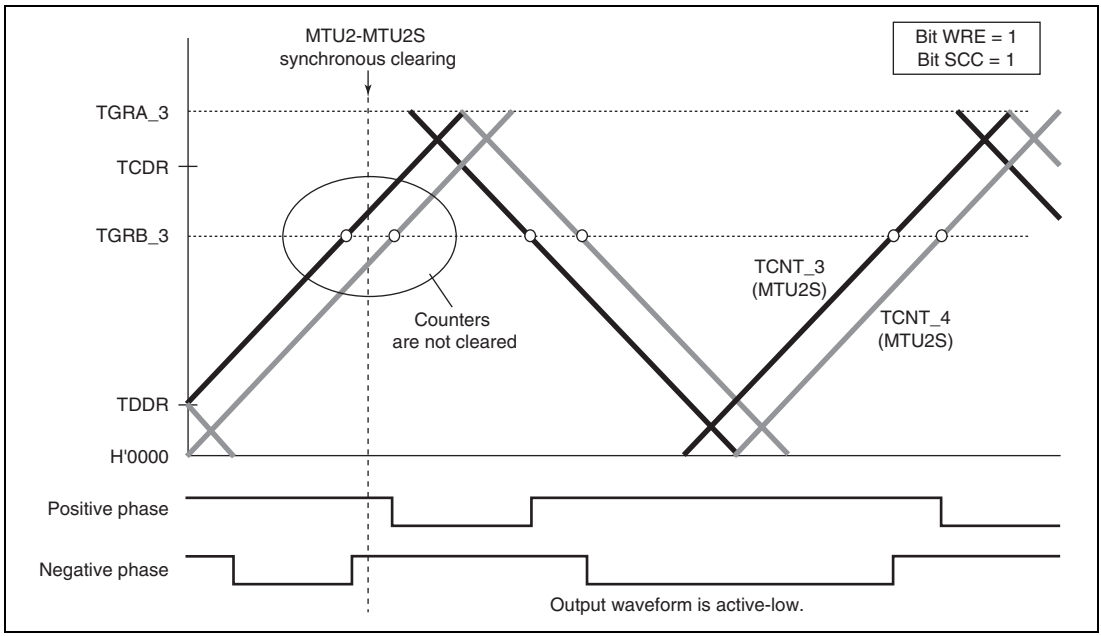


Figure 10.64 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

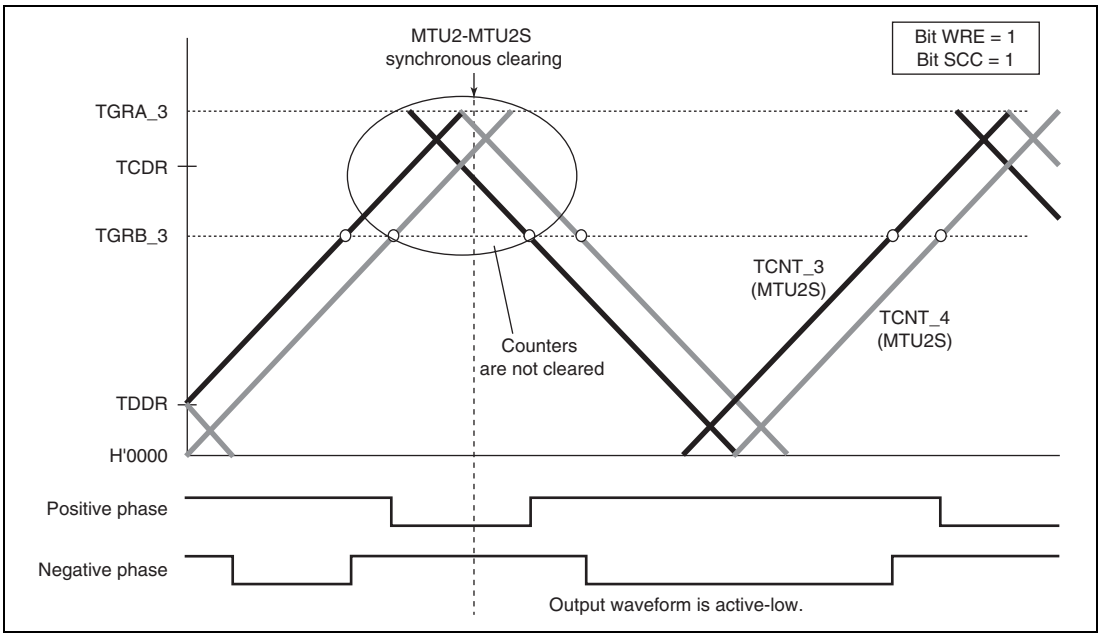


Figure 10.65 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

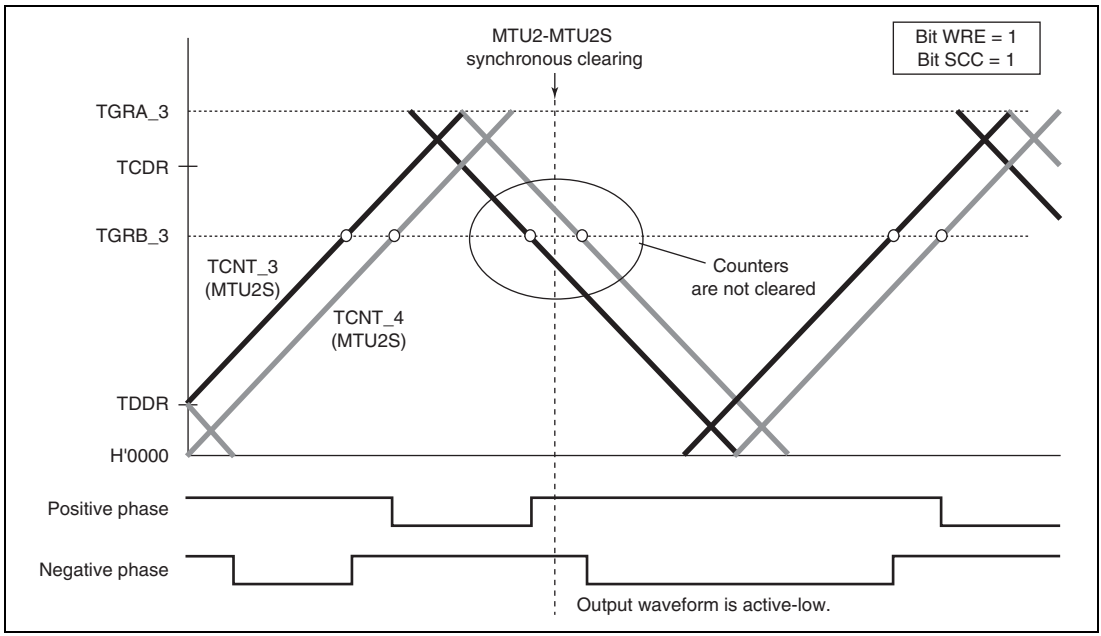


Figure 10.66 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

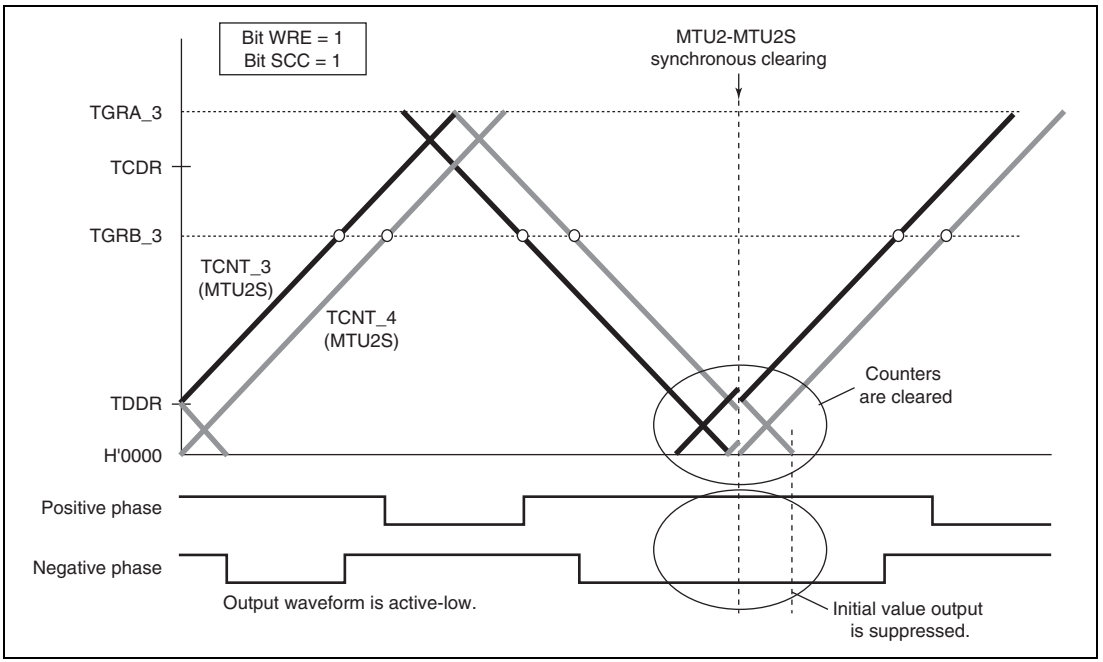


Figure 10.67 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 10.56; Bit WRE is 1 and Bit SCC is 1 in TWCR of MTU2S)

(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 10.68 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.

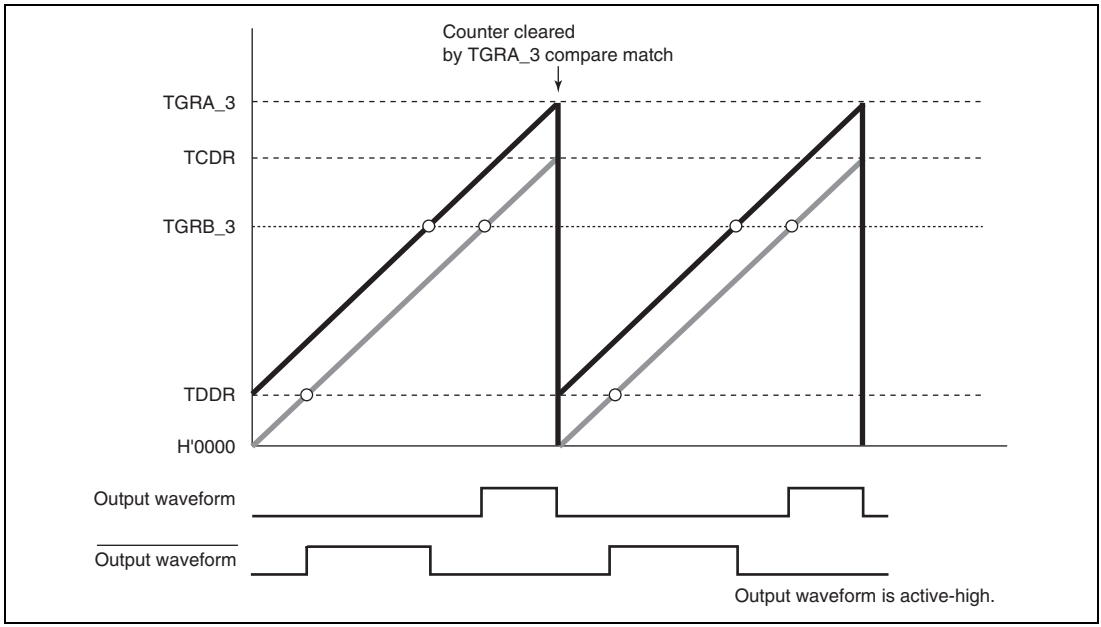


Figure 10.68 Example of Counter Clearing Operation by TGRA_3 Compare Match

(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 10.69 to 10.72 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.

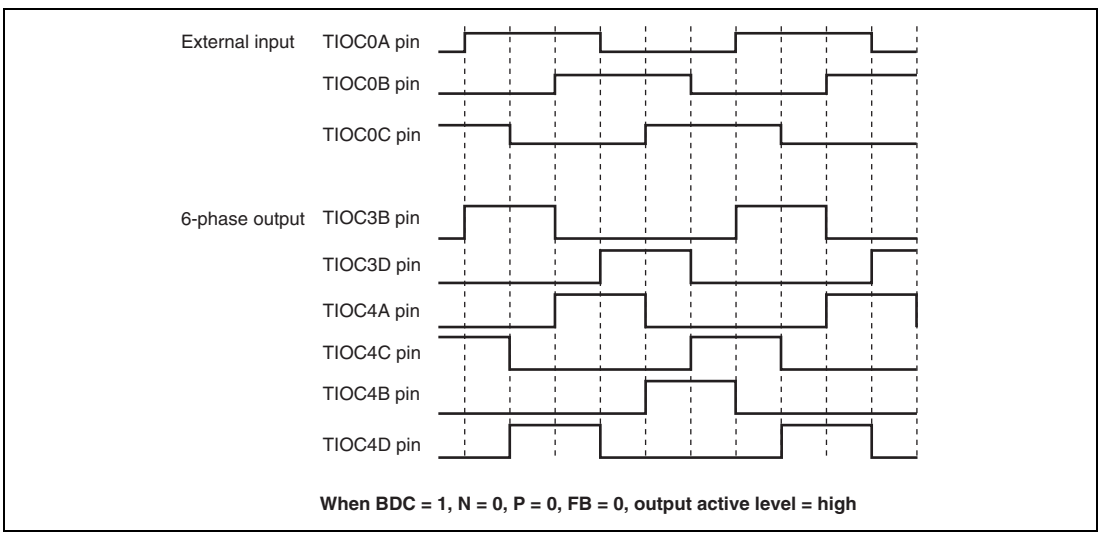


Figure 10.69 Example of Output Phase Switching by External Input (1)

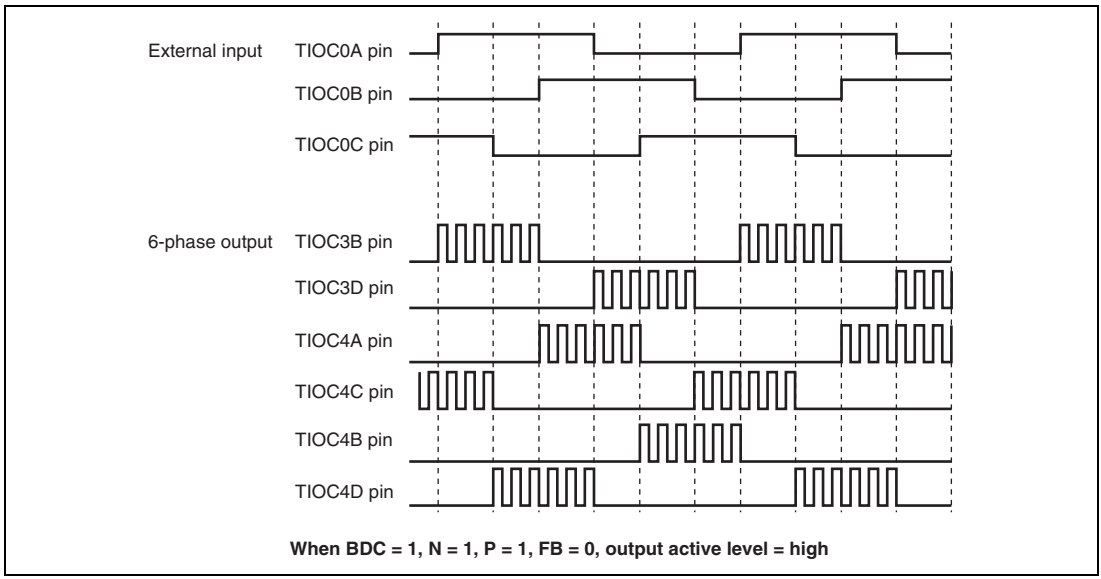


Figure 10.70 Example of Output Phase Switching by External Input (2)

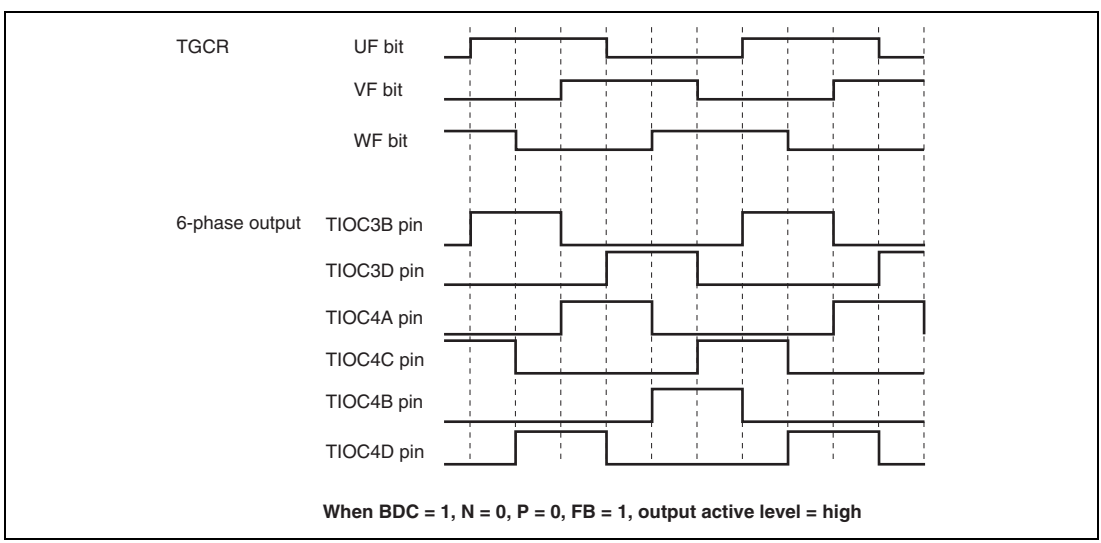


Figure 10.71 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

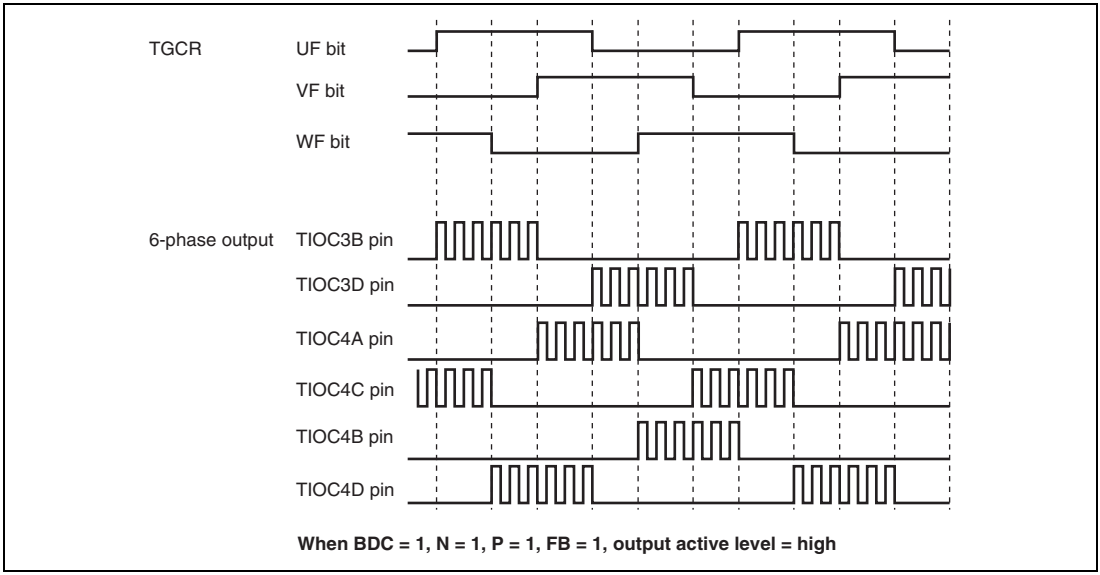


Figure 10.72 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(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 10.4.9, 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 10.73 shows an example of the interrupt skipping operation setting procedure. Figure 10.74 shows the periods during which interrupt skipping count can be changed.

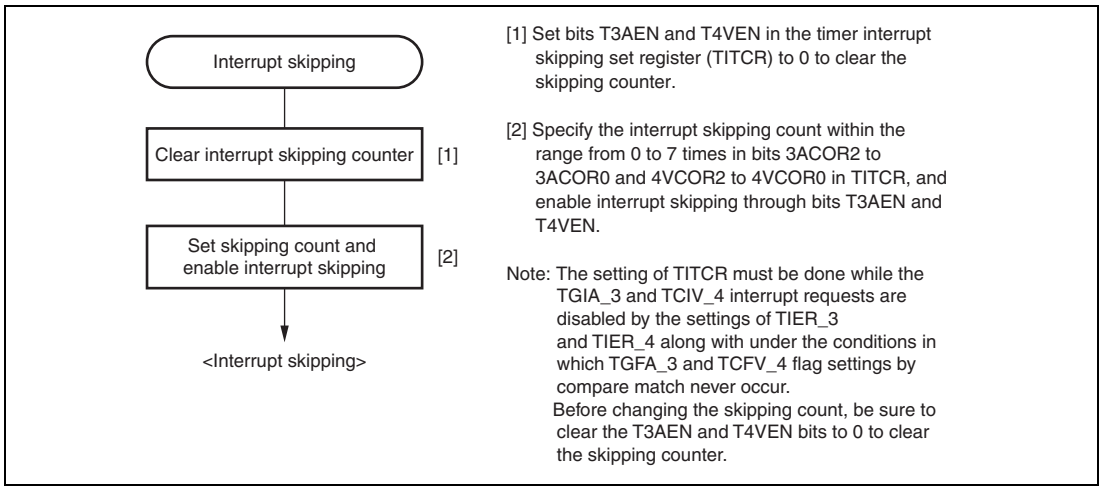


Figure 10.73 Example of Interrupt Skipping Operation Setting Procedure

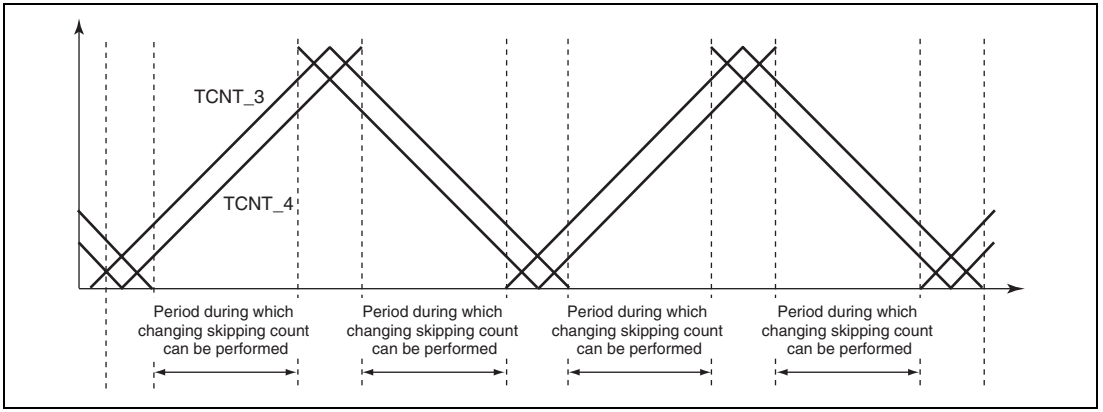


Figure 10.74 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 10.75 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).

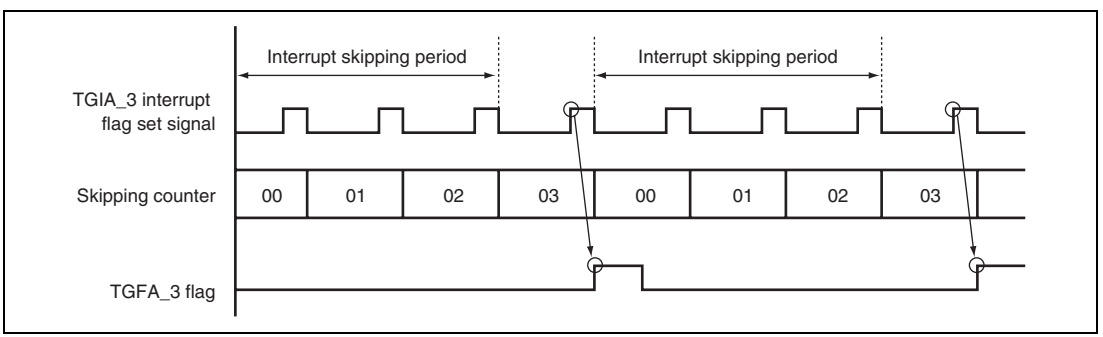


Figure 10.75 Example of Interrupt Skipping Operation

(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).

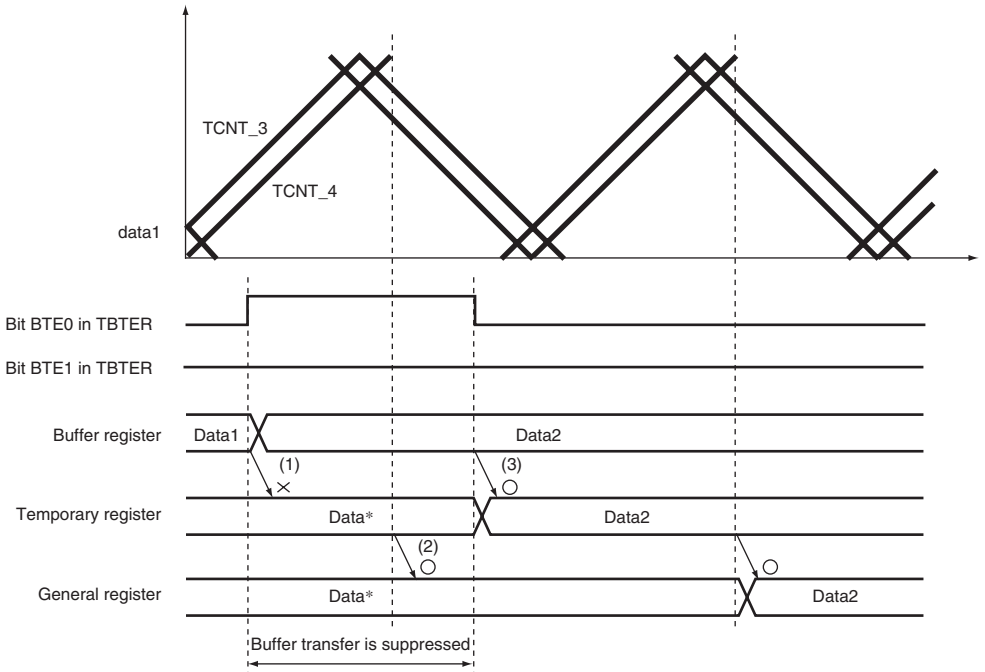
Figure 10.76 shows an example of operation when buffer transfer is suppressed (BTE1 = 0 and BTE0 = 1). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 10.77 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.

There are two types of timing in which data is transferred from the buffer register to the temporary register or to general register, depending on the buffer register modification timing after an interrupt occurrence.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 10.78 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]

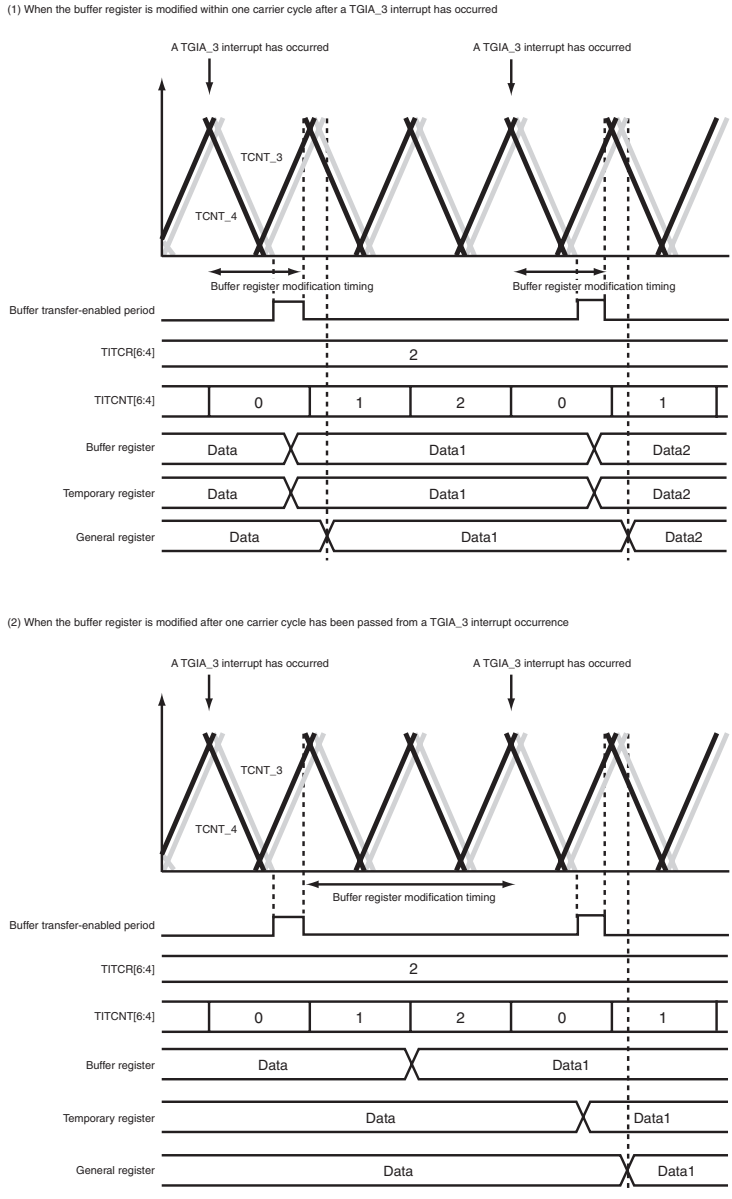
(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).

(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.

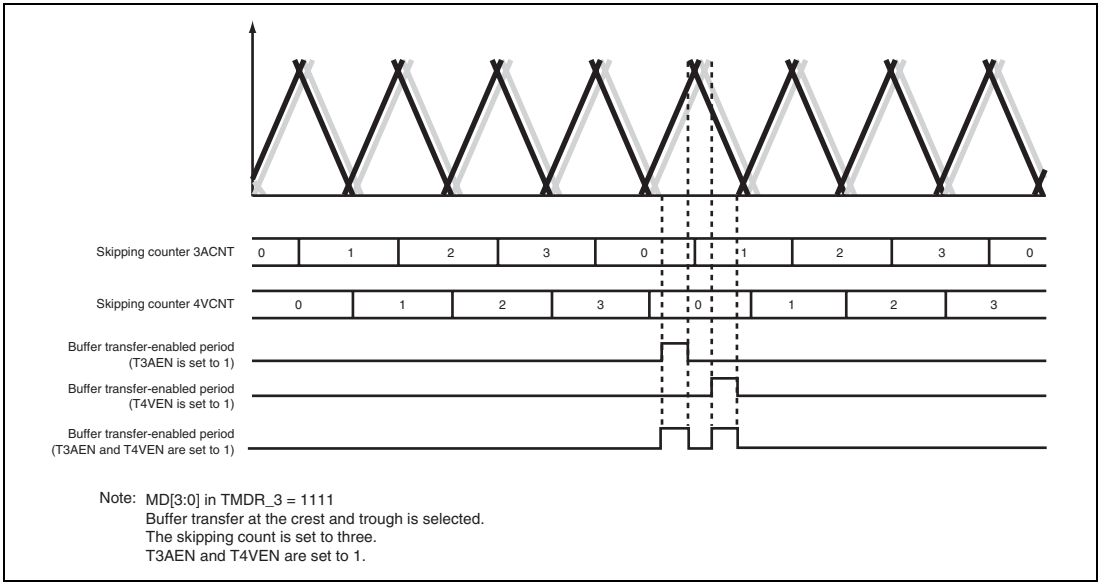
Note: * When buffer transfer at the crest is selected.

Figure 10.76 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



Note: MD[3:0] in TMDR_3 = 1101
Buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and cleared to 0, respectively.

Figure 10.77 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)



Note: MD[3:0] in TMDR_3 = 1111
Buffer transfer at the crest and trough is selected.
The skipping count is set to three.
T3AEN and T4VEN are set to 1.

Figure 10.78 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 12, Port Output Enable 2 (POE2), for details.

10.4.9 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 10.79 shows an example of procedure for specifying the A/D converter start request delaying function.

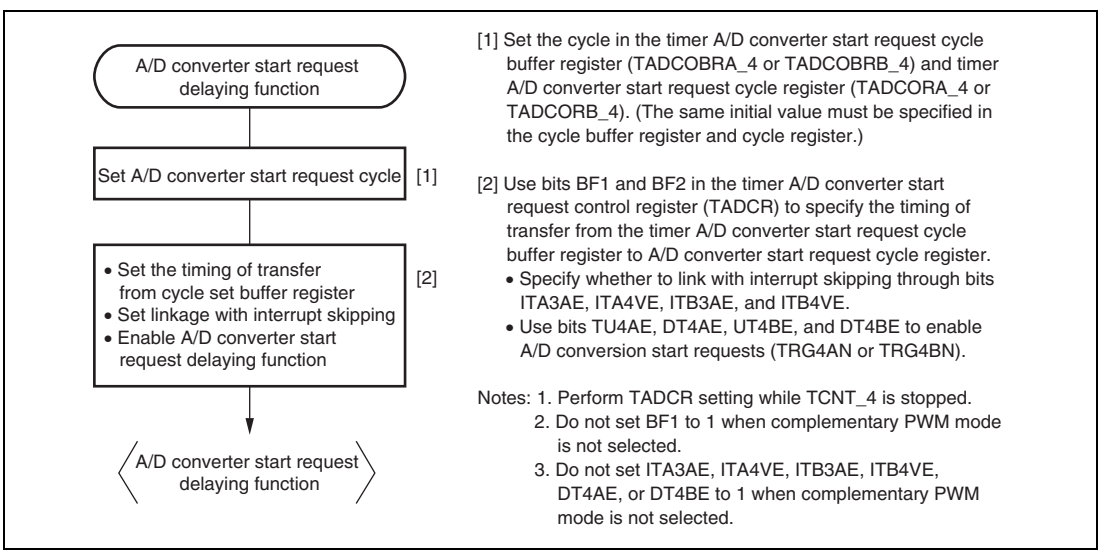


Figure 10.79 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

- Basic Operation Example of A/D Converter Start Request Delaying Function

  Figure 10.80 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.

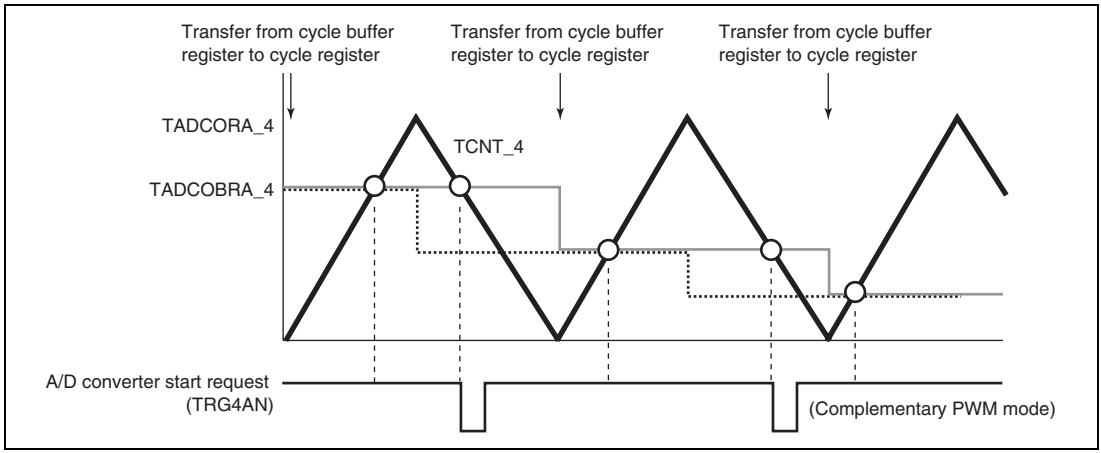


Figure 10.80 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

- 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 10.81 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 10.82 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).

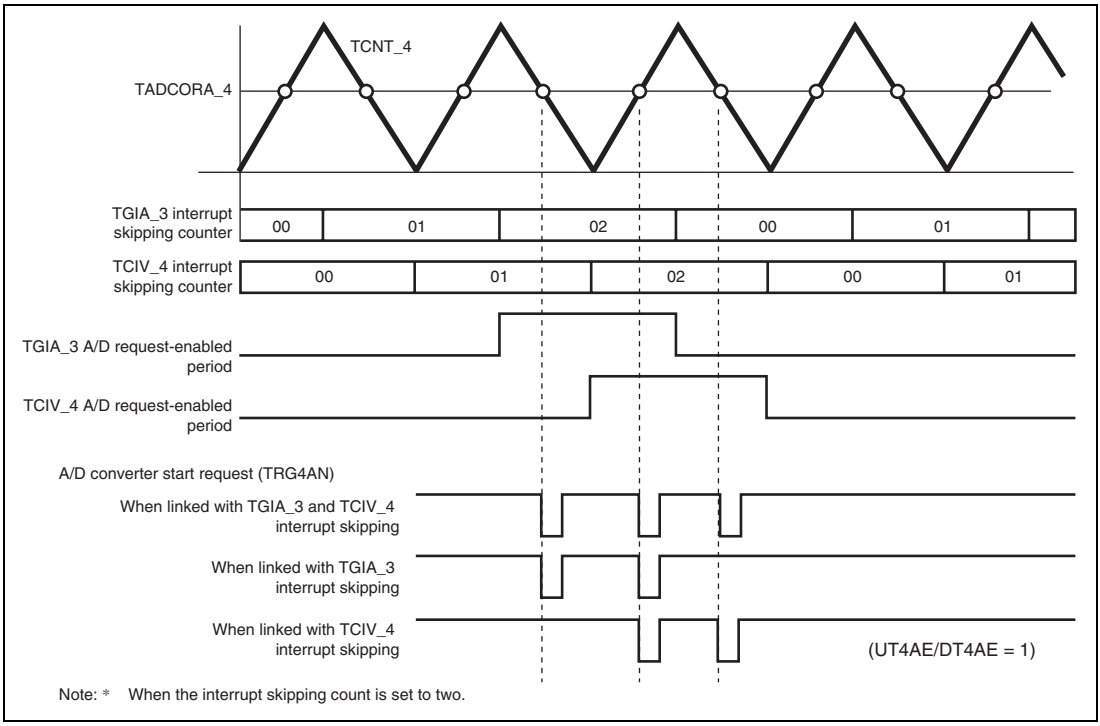


Figure 10.81 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

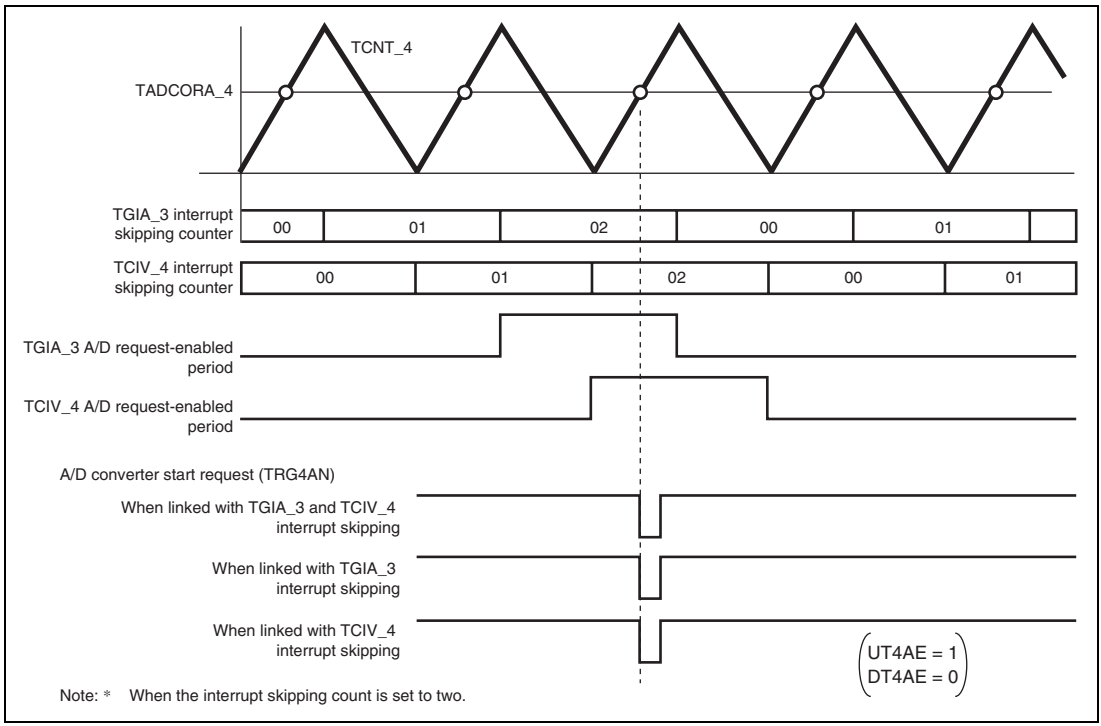


Figure 10.82 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

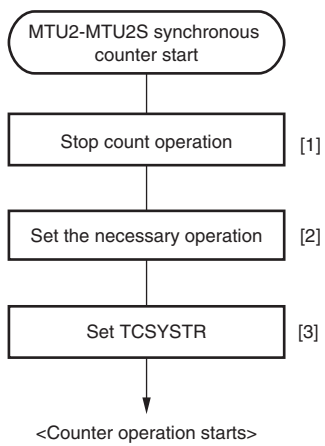
10.4.10 MTU2–MTU2S Synchronous Operation

(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 10.83 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 10.4.7, Reset-Synchronized PWM Mode, and section 10.4.8, Complementary PWM Mode.

Figure 10.83 Example of Synchronous Counter Start Setting Procedure

(b) Examples of Synchronous Counter Start Operation

Figures 10.84 (1) to (4) show examples of synchronous counter start operation when the clock frequency ratios between the MTU2 and MTU2S are 1:1, 1:2, 1:3, and 1:4, respectively. In these examples, the count clock is set to Pϕ/1.

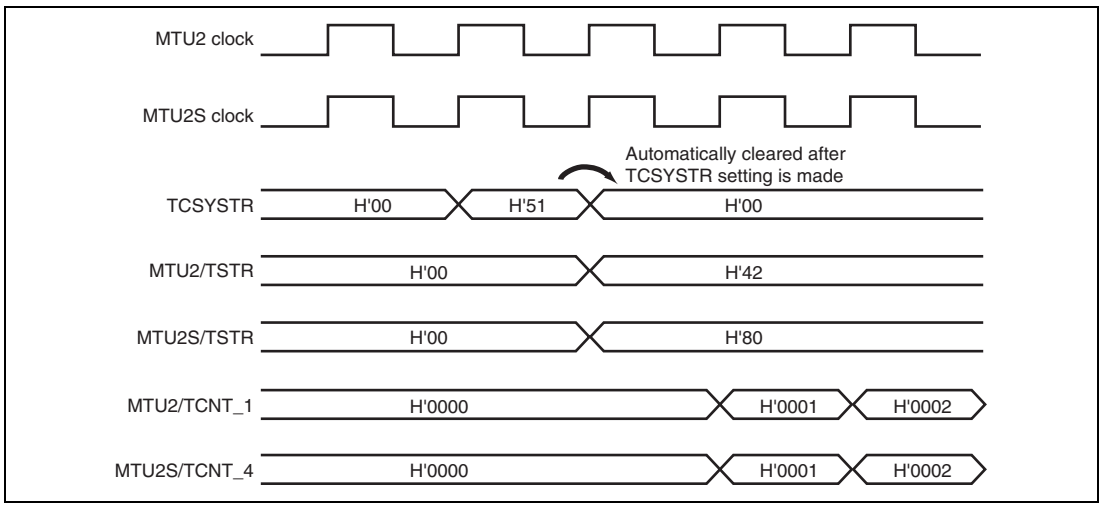


Figure 10.84 (1) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:1)

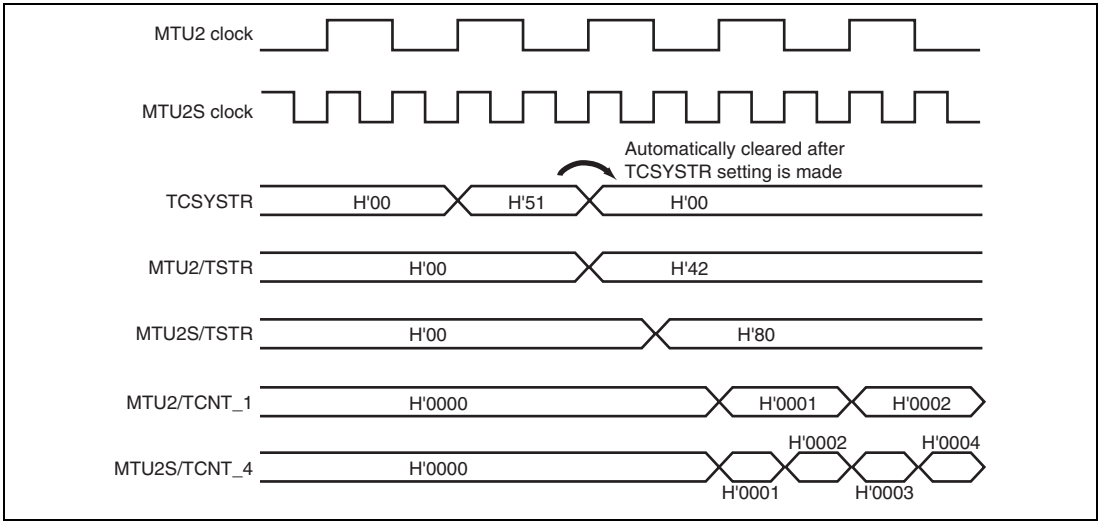


Figure 10.84 (2) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:2)

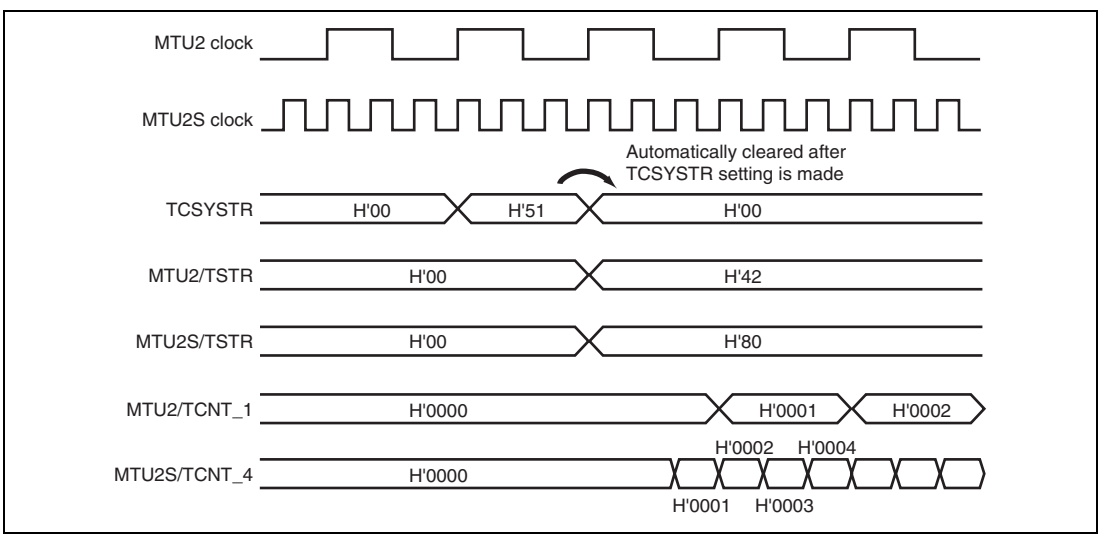


Figure 10.84 (3) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:3)

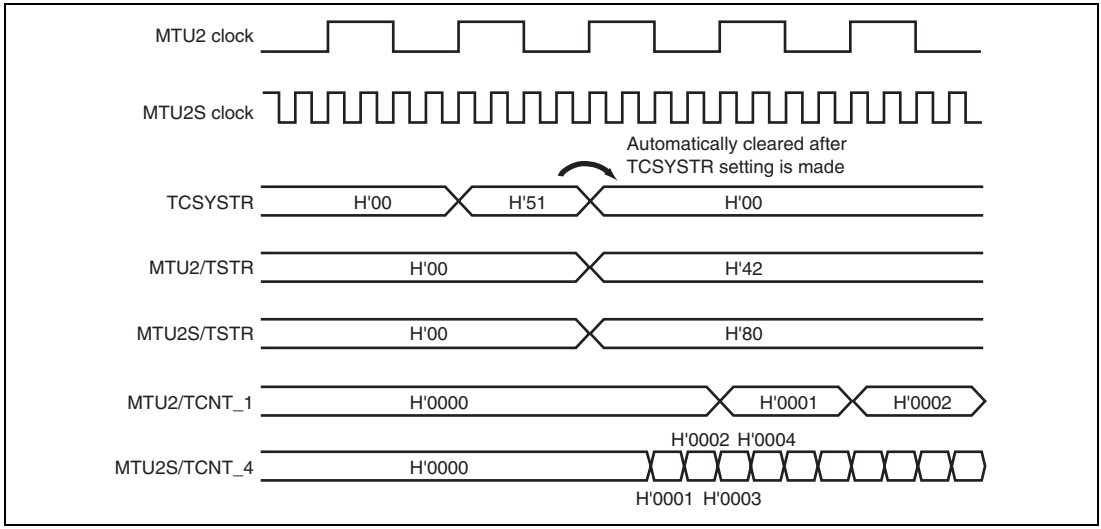


Figure 10.84 (4) Example of Synchronous Counter Start Operation (MTU2-to-MTU2S Clock Frequency Ratio = 1:4)

(2) MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (MTU2–MTU2S Synchronous Counter Clearing)

The MTU2S counters can be cleared by sources for setting the flags in TSR_0 to TSR_2 in the MTU2 through the TSYCR_3 settings in the MTU2S.

(a) Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

Figure 10.85 shows an example of procedure for specifying MTU2S counter clearing by MTU2 flag setting source.

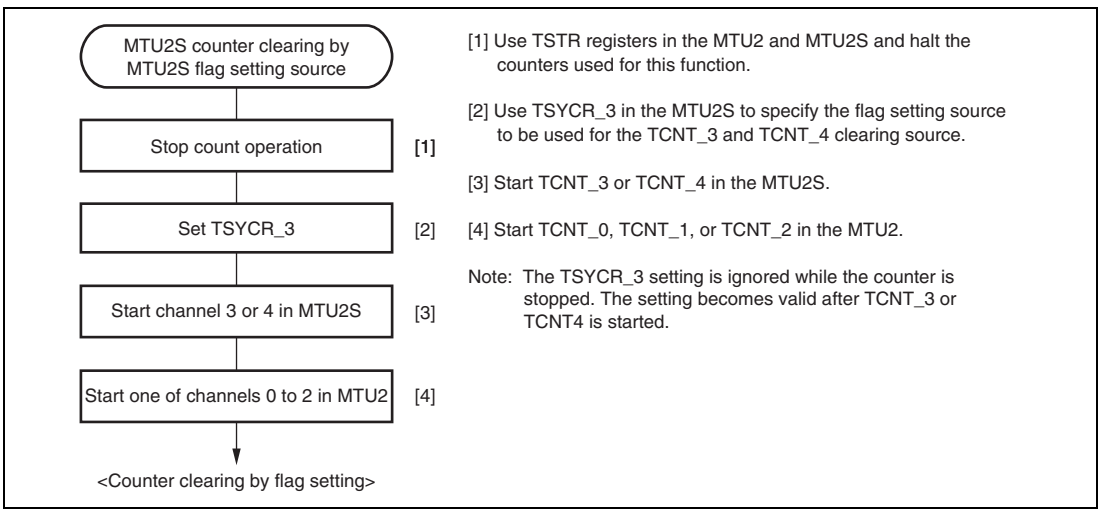


Figure 10.85 Example of Procedure for Specifying MTU2S Counter Clearing by MTU2 Flag Setting Source

(b) Examples of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source

Figures 10.86 (1) and 10.86 (2) show examples of MTS2S counter clearing caused by MTU2 flag setting source.

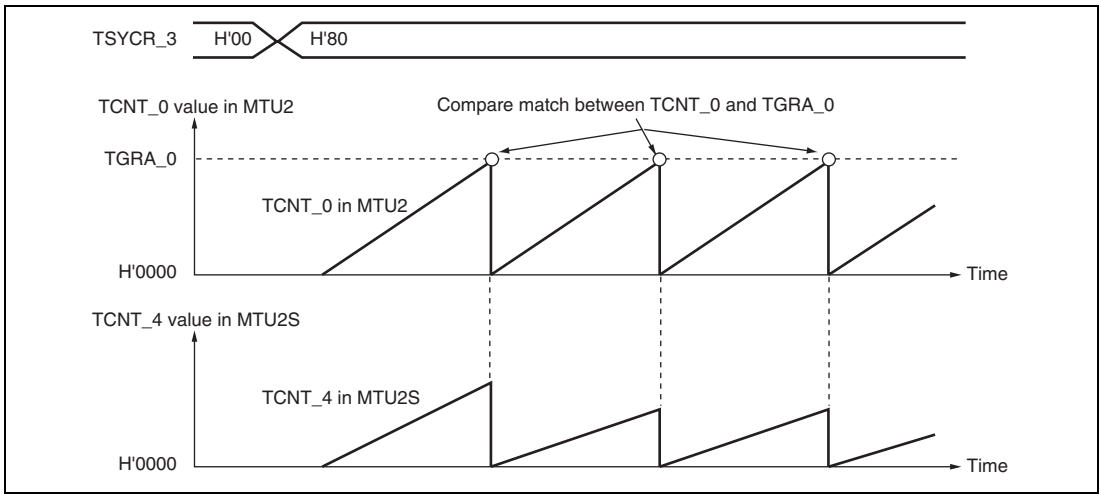


Figure 10.86 (1) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (1)

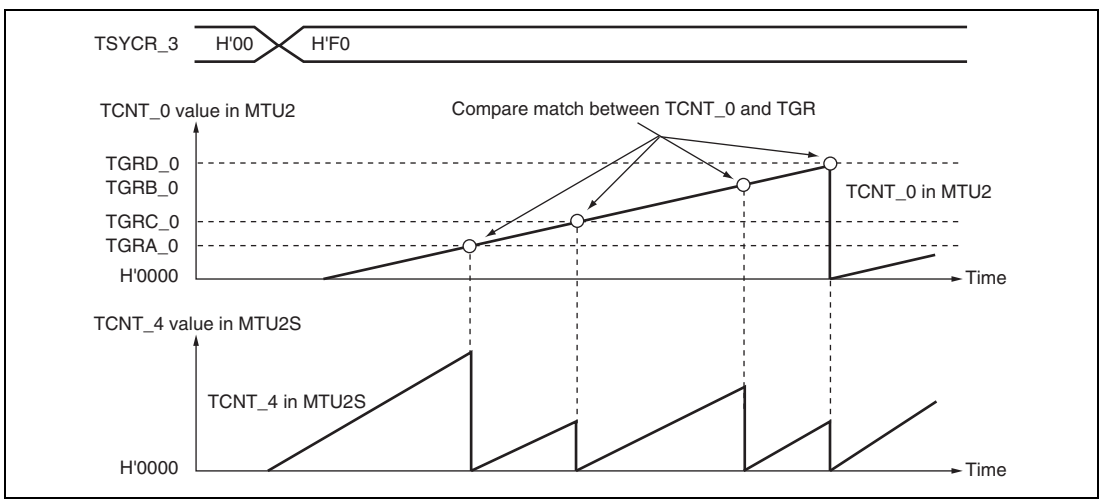


Figure 10.86 (2) Example of MTU2S Counter Clearing Caused by MTU2 Flag Setting Source (2)

10.4.11 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

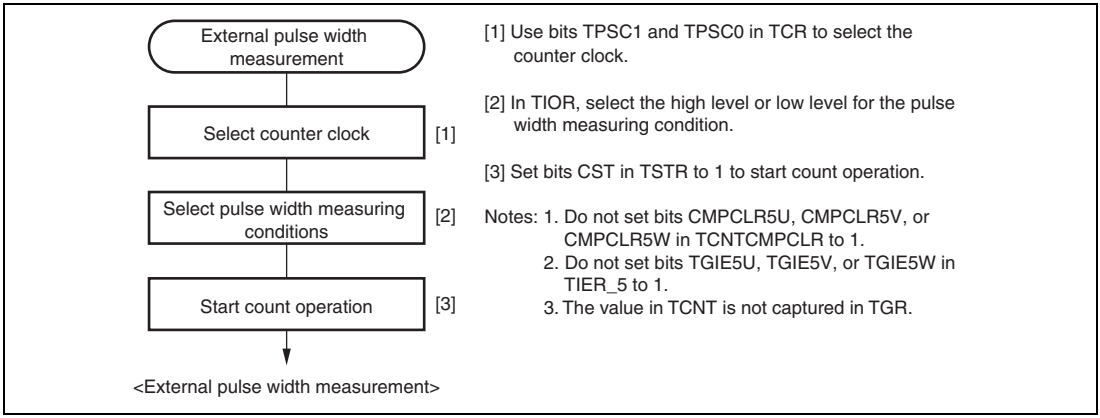


Figure 10.87 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

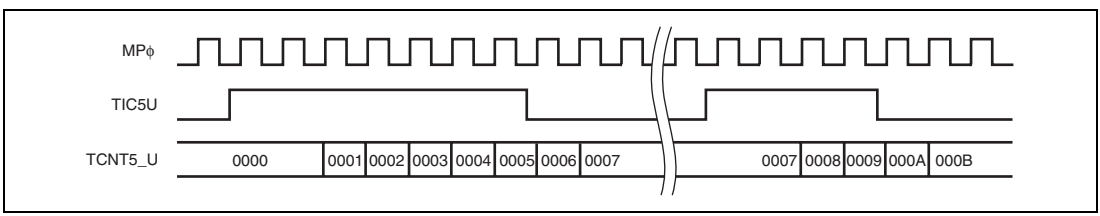


Figure 10.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)

10.4.12 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.

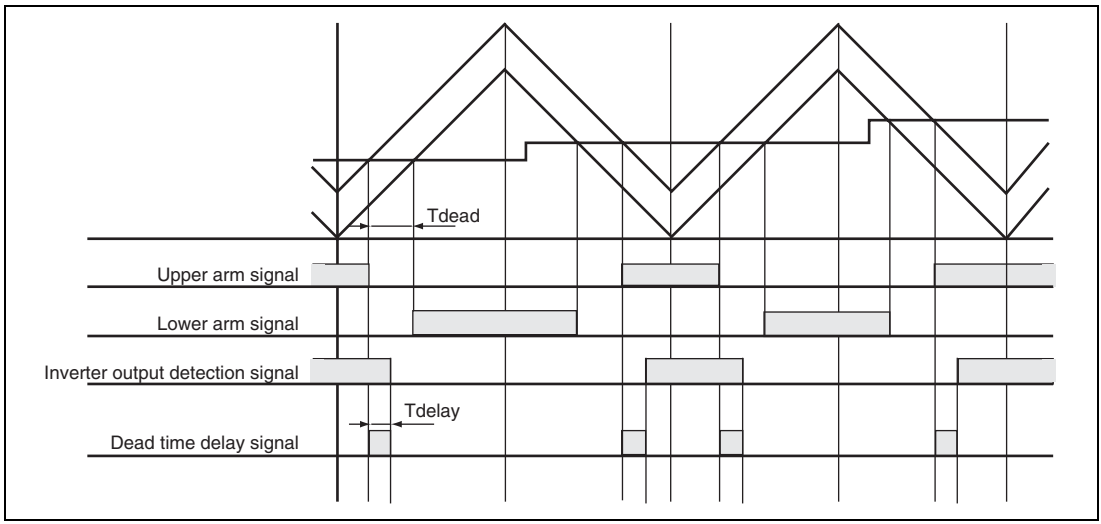
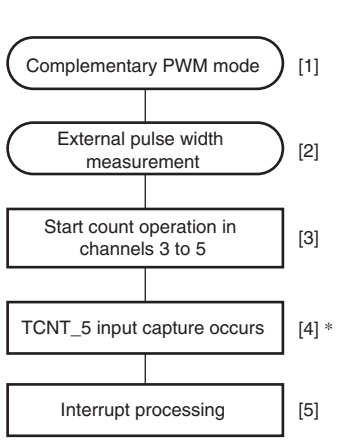


Figure 10.89 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 10.90 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 10.4.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in channel 5. For details, refer to section 10.4.11, 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 10.90 Example of Dead Time Compensation Setting Procedure

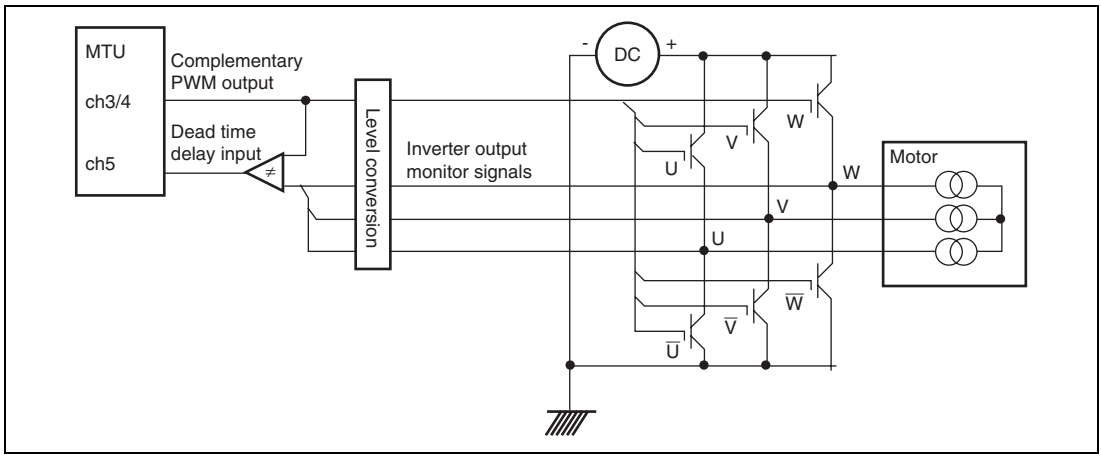


Figure 10.91 Example of Motor Control Circuit Configuration

10.4.13 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 10.92 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).

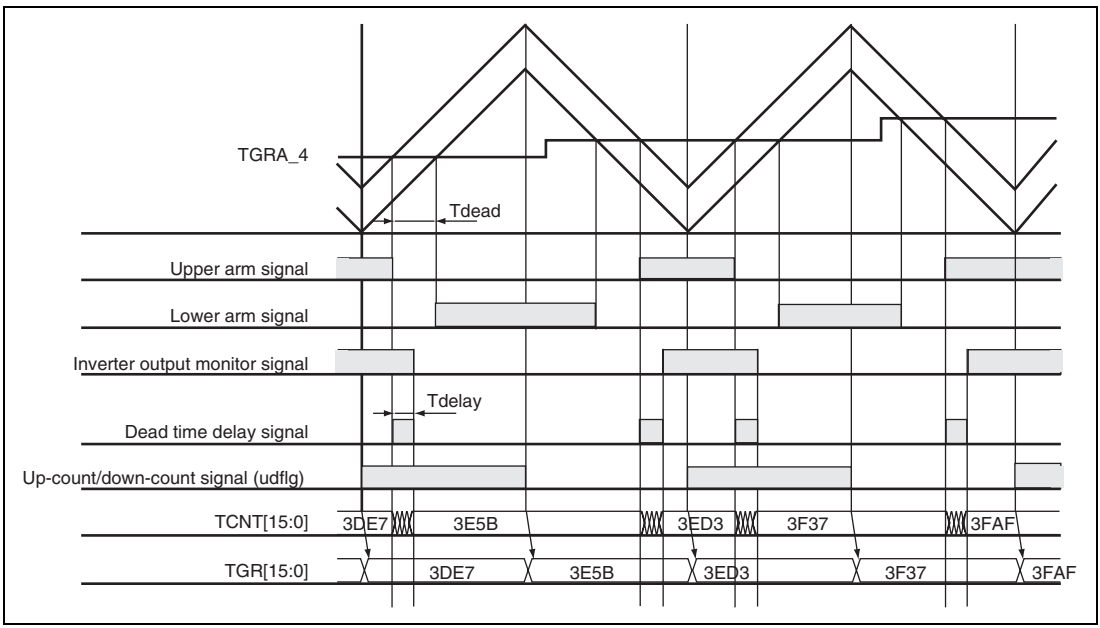


Figure 10.92 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

10.5 Interrupt Sources

10.5.1 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 10.57 lists the MTU2 interrupt sources.

Table 10.57 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.

10.5.2 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 9, 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.

10.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 10.58 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 TAD4AE or TAD4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 10.4.9, 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 10.58 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

10.6 Operation Timing

10.6.1 Input/Output Timing

(1) TCNT Count Timing

Figures 10.93 and 94 show TCNT count timing in internal clock operation, and figure 10.95 shows TCNT count timing in external clock operation (normal mode), and figure 10.96 shows TCNT count timing in external clock operation (phase counting mode).

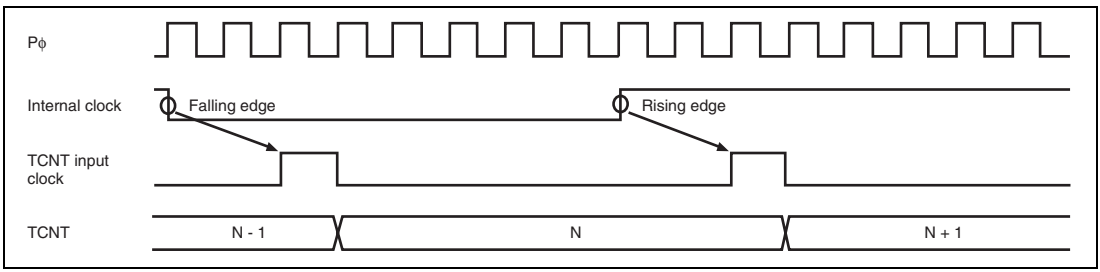


Figure 10.93 Count Timing in Internal Clock Operation (Channels 0 to 4)

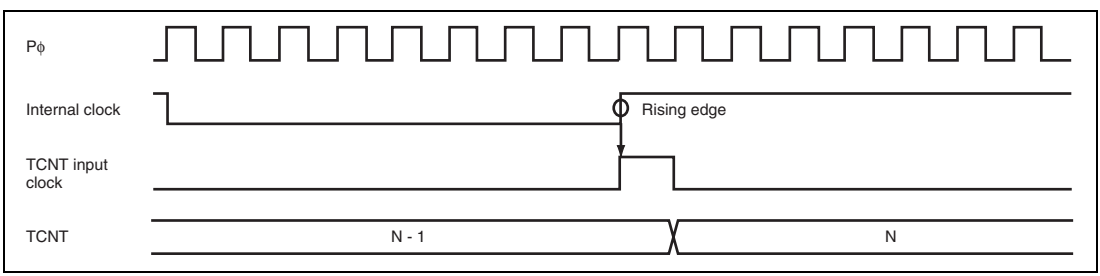


Figure 10.94 Count Timing in Internal Clock Operation (Channel 5)

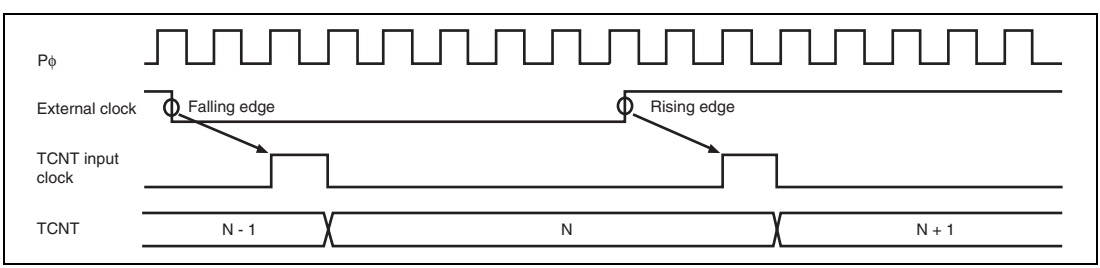


Figure 10.95 Count Timing in External Clock Operation (Channels 0 to 4)

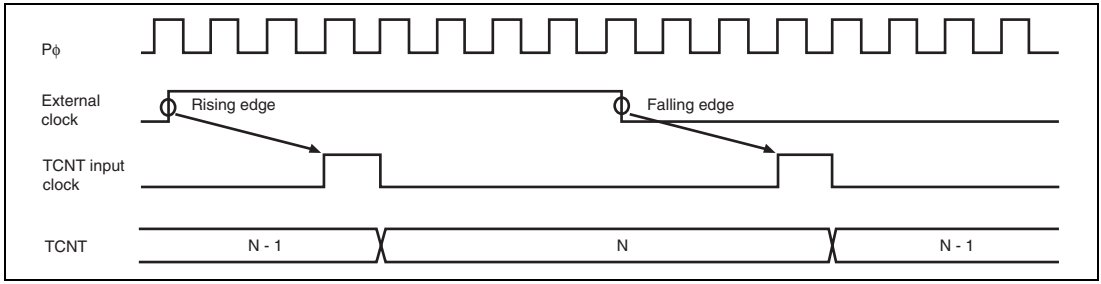


Figure 10.96 Count Timing in External Clock Operation (Phase Counting Mode)

(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 10.97 shows output compare output timing (normal mode and PWM mode) and figure 10.98 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

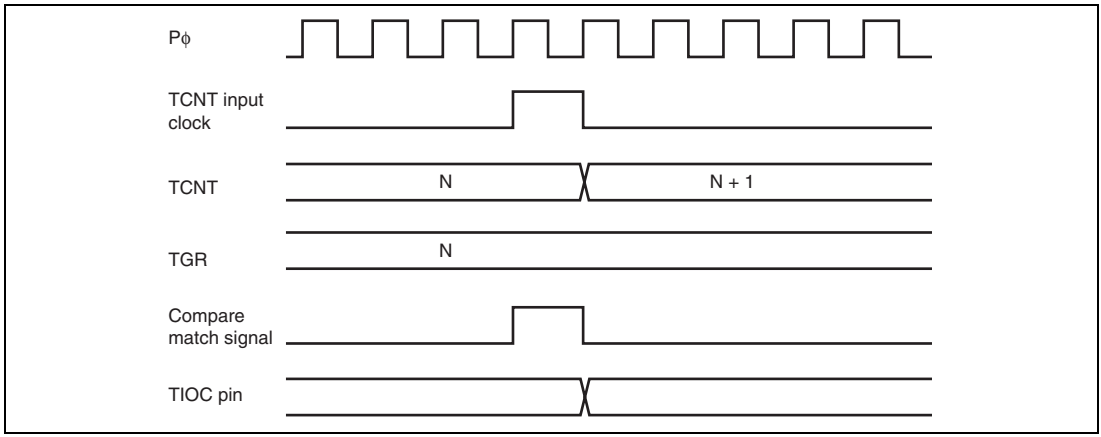


Figure 10.97 Output Compare Output Timing (Normal Mode/PWM Mode)

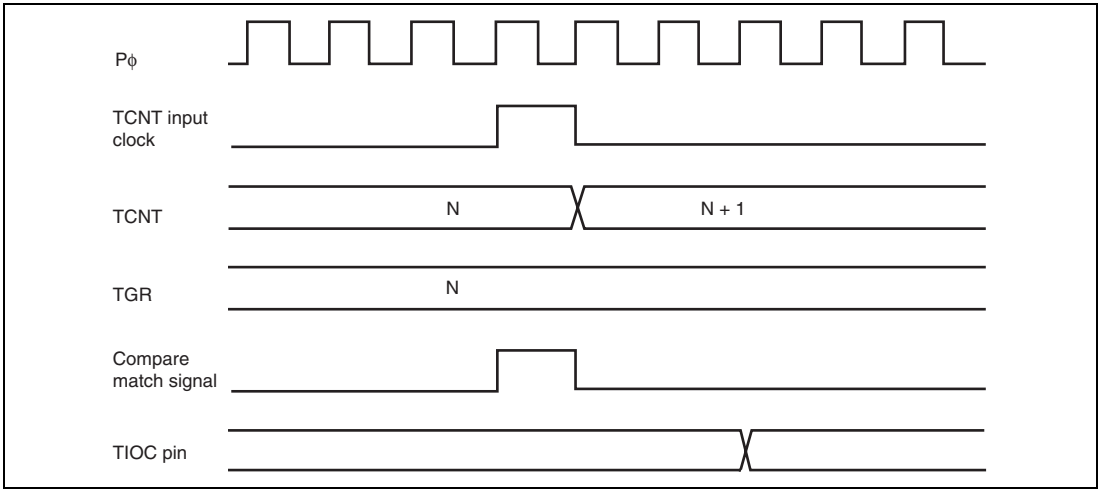


Figure 10.98 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 10.99 shows input capture signal timing.

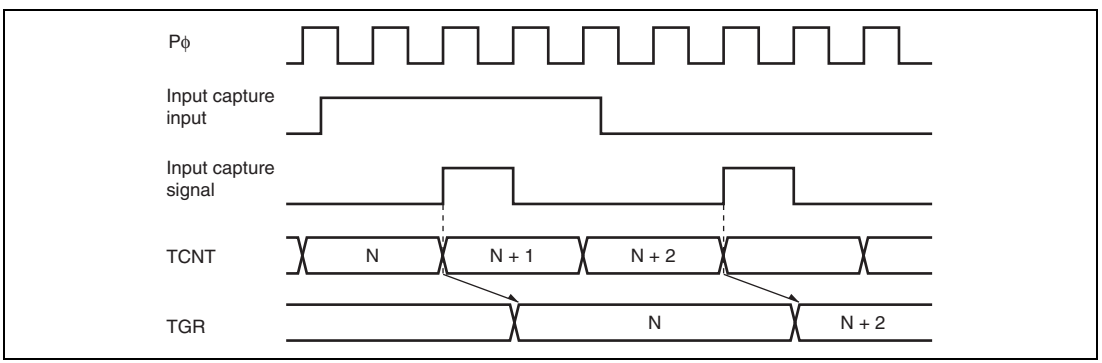


Figure 10.99 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figures 10.100 and 101 show the timing when counter clearing on compare match is specified, and figure 10.102 shows the timing when counter clearing on input capture is specified.

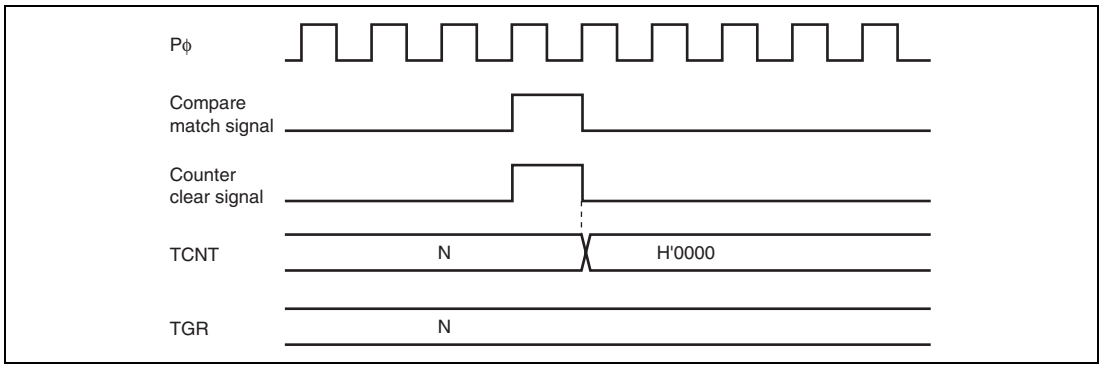


Figure 10.100 Counter Clear Timing (Compare Match) (Channels 0 to 4)

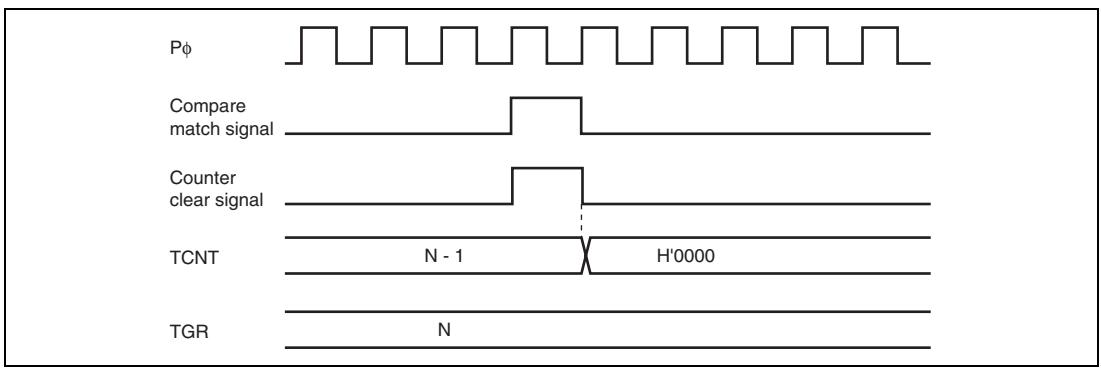


Figure 10.101 Counter Clear Timing (Compare Match) (Channel 5)

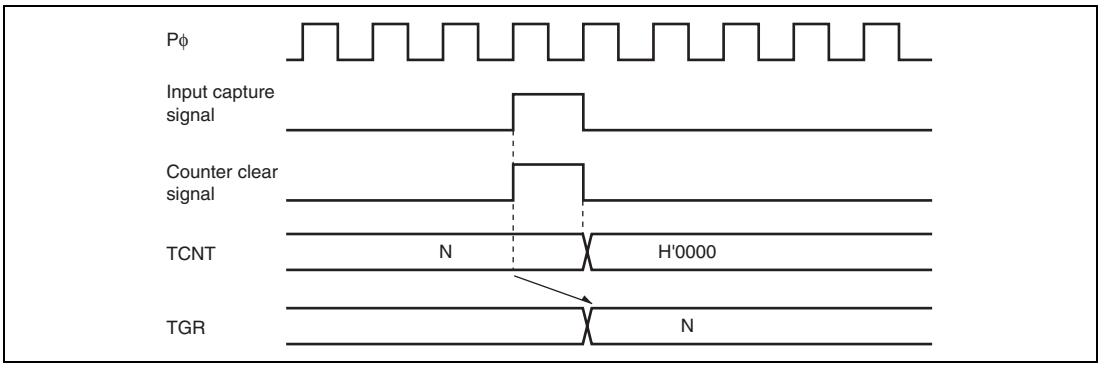


Figure 10.102 Counter Clear Timing (Input Capture) (Channels 0 to 5)

(5) Buffer Operation Timing

Figures 10.103 to 10.105 show the timing in buffer operation.

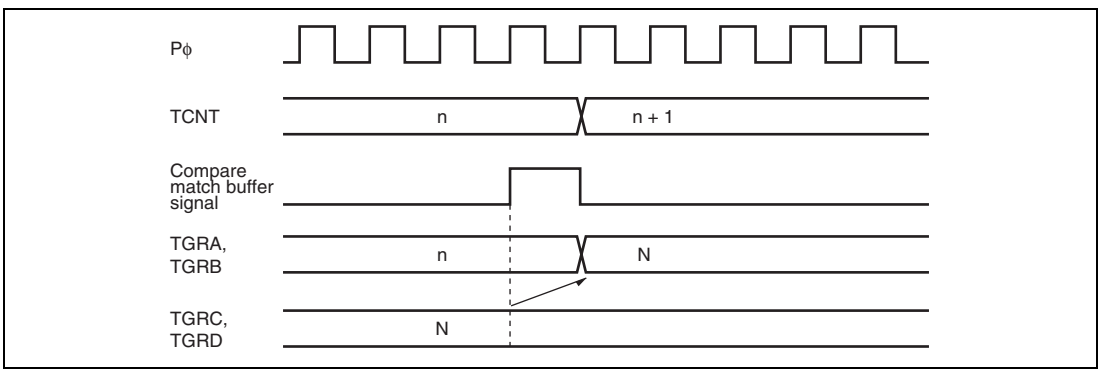


Figure 10.103 Buffer Operation Timing (Compare Match)

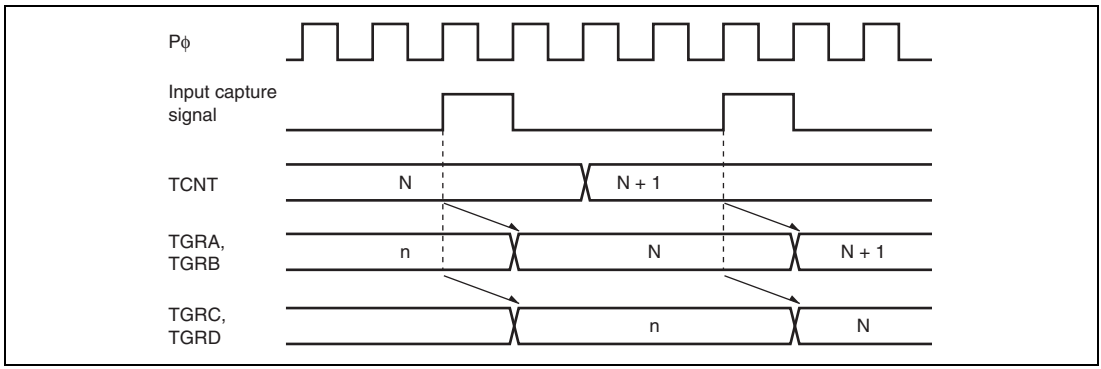


Figure 10.104 Buffer Operation Timing (Input Capture)

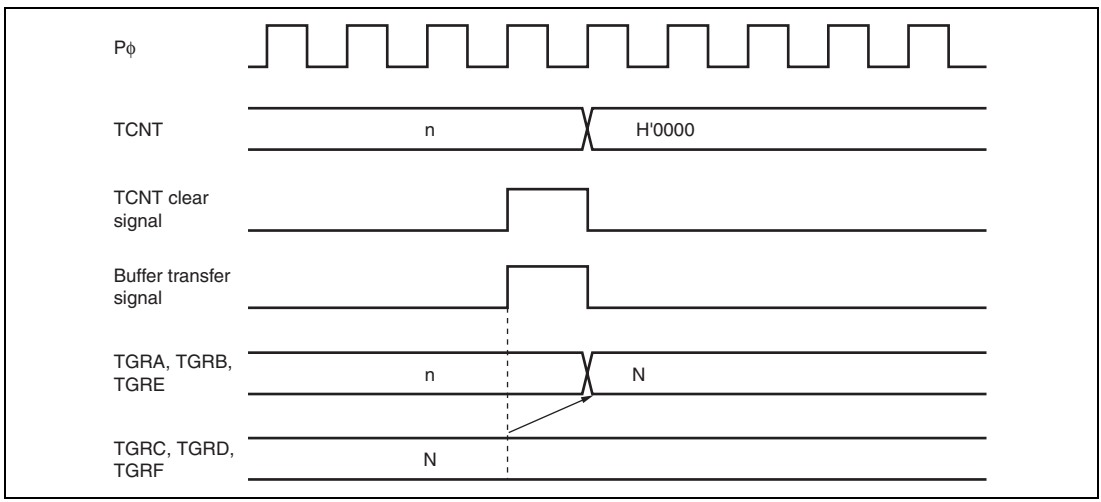


Figure 10.105 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 10.106 to 10.108 show the buffer transfer timing in complementary PWM mode.

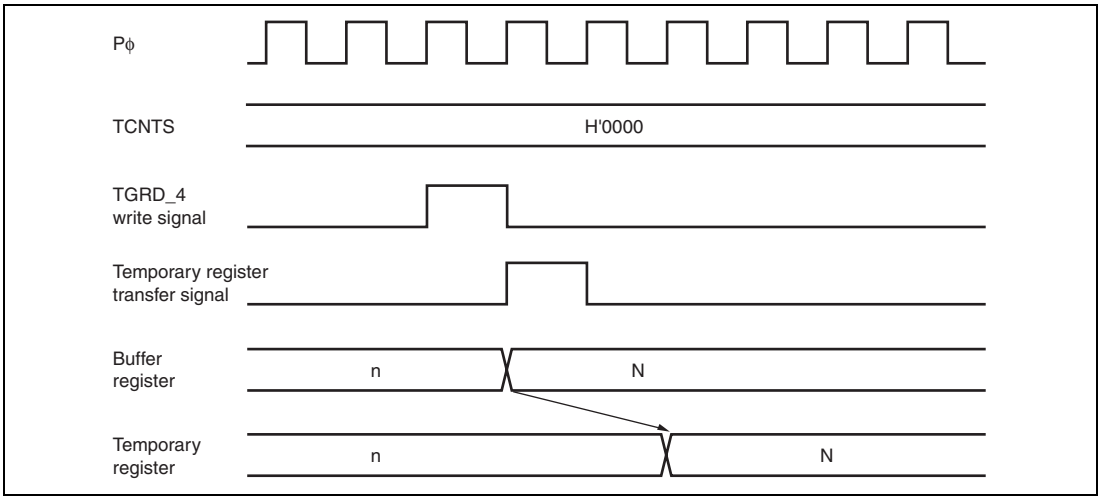


Figure 10.106 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

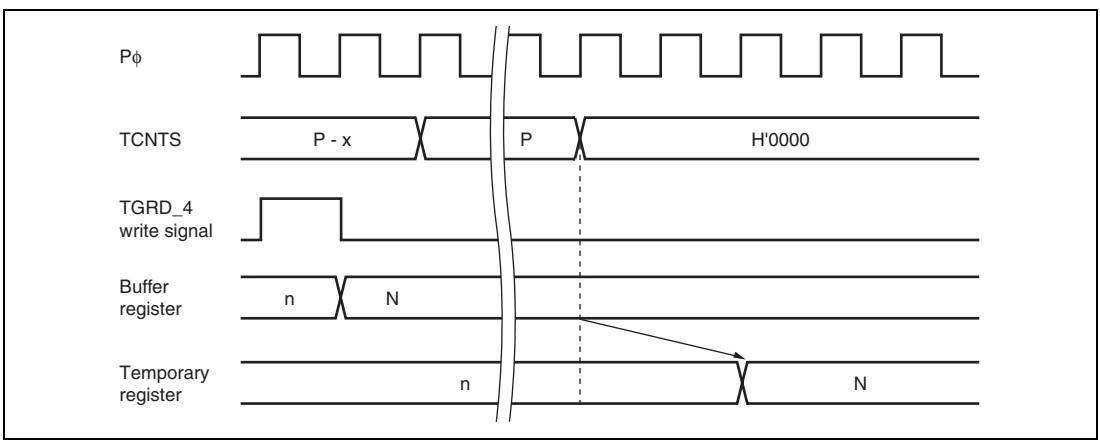


Figure 10.107 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

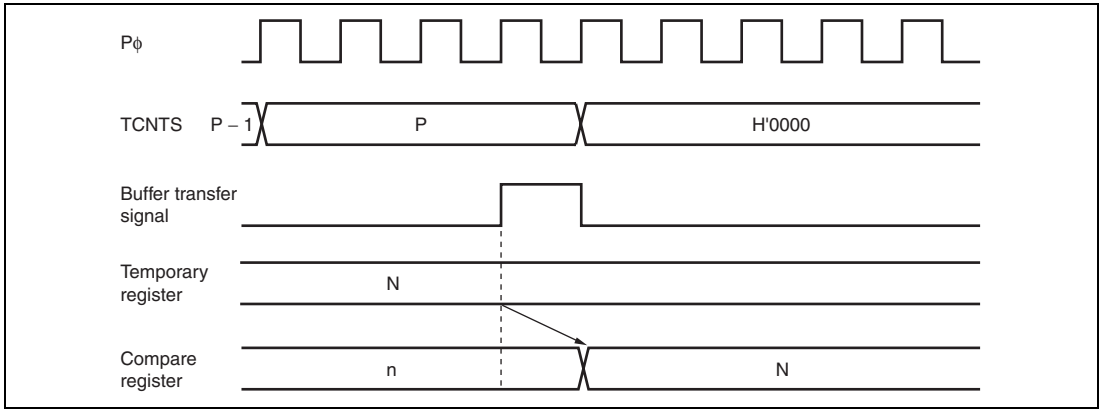


Figure 10.108 Transfer Timing from Temporary Register to Compare Register

10.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figures 10.109 and 110 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

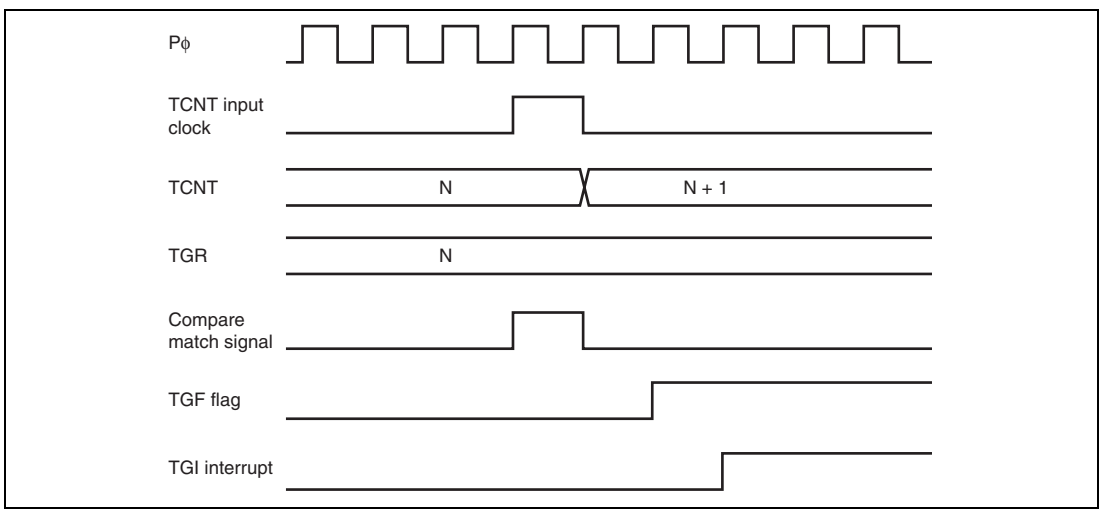


Figure 10.109 TGI Interrupt Timing (Compare Match)

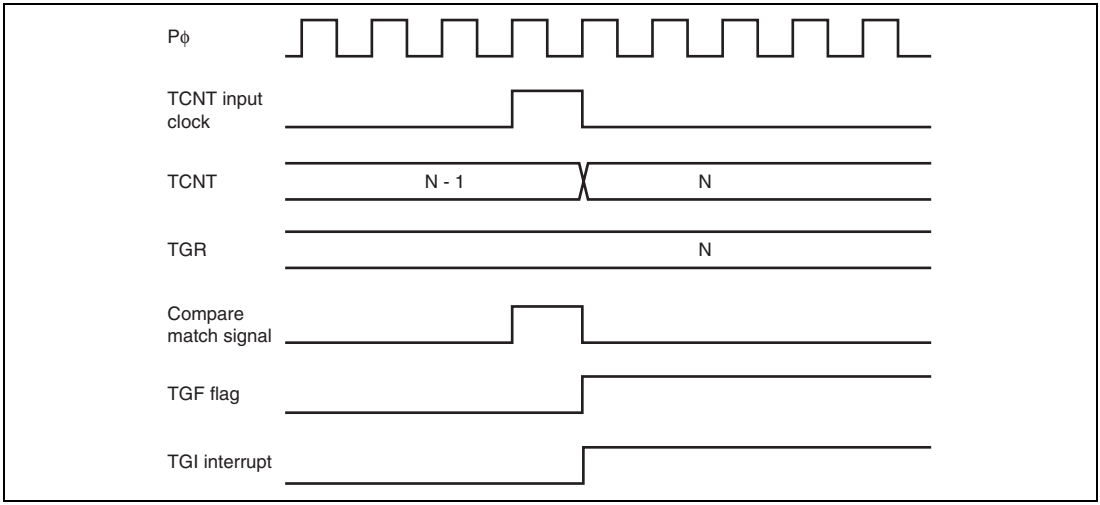


Figure 10.110 TGI Interrupt Timing (Compare Match) (Channel 5)

(2) TGF Flag Setting Timing in Case of Input Capture

Figures 10.111 and 112 show the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

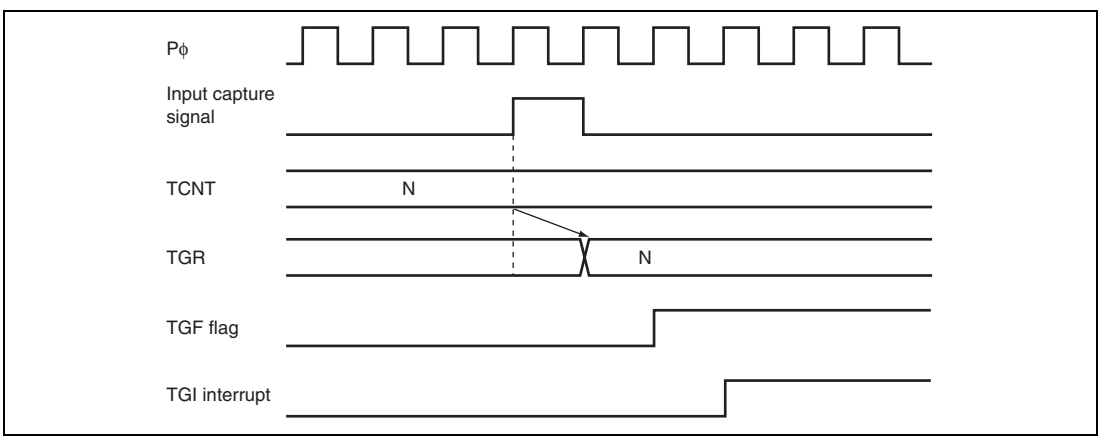


Figure 10.111 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)

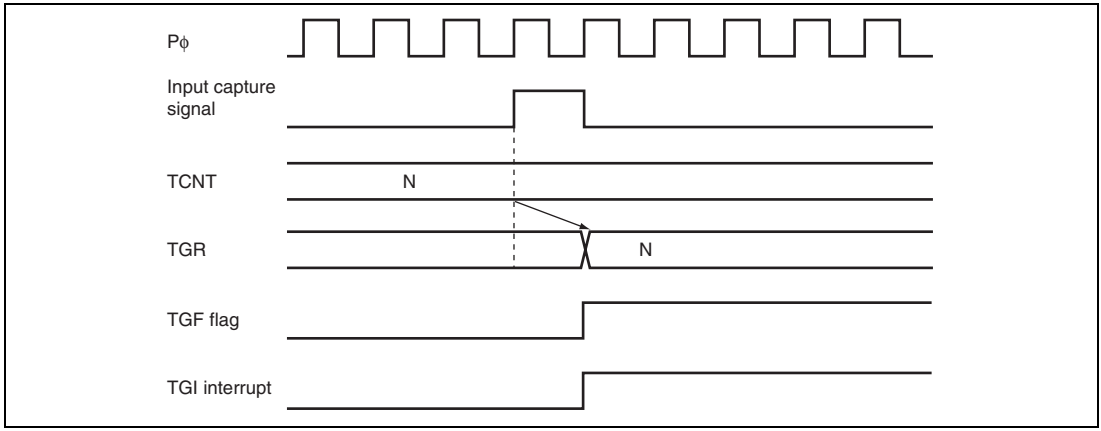


Figure 10.112 TGI Interrupt Timing (Input Capture) (Channel 5)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 10.113 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 10.114 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

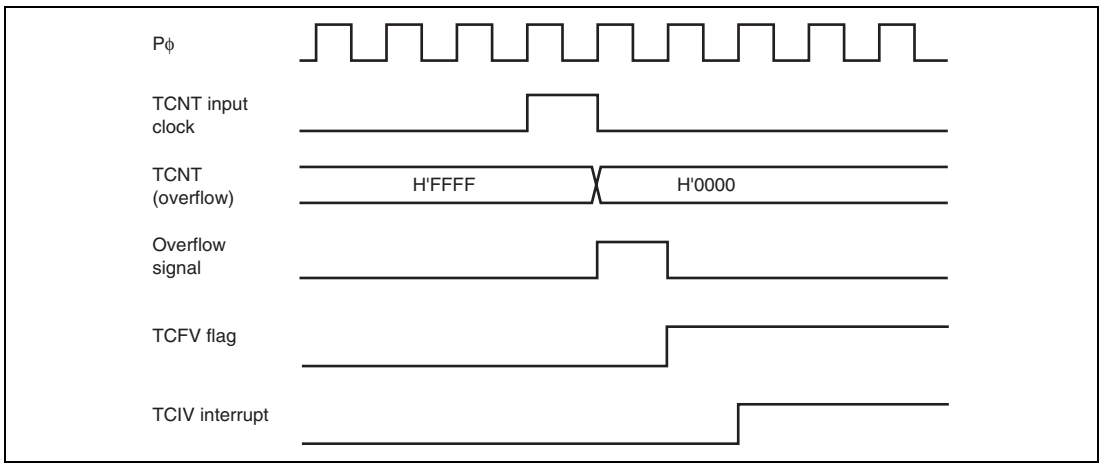


Figure 10.113 TCIV Interrupt Setting Timing

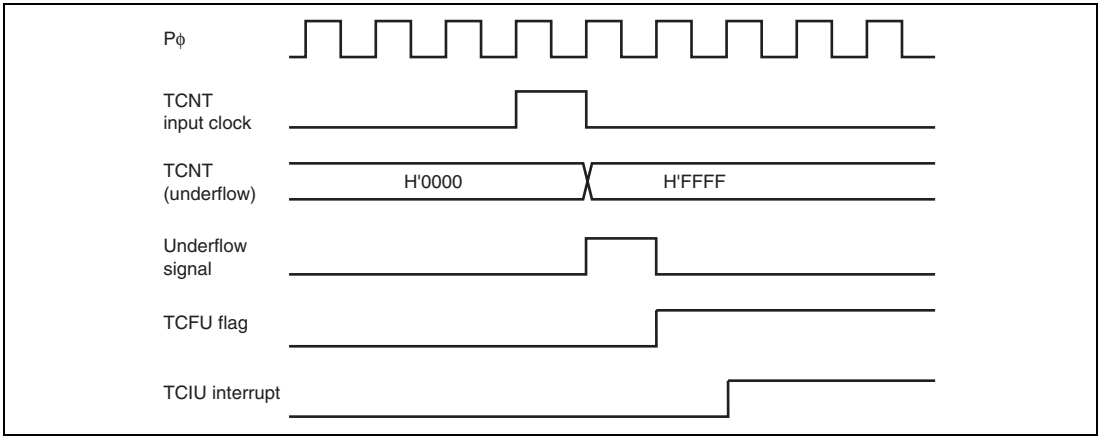


Figure 10.114 TCIU Interrupt Setting Timing

(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 10.115 and 116 show the timing for status flag clearing by the CPU, and figure 10.117 shows the timing for status flag clearing by the DMAC.

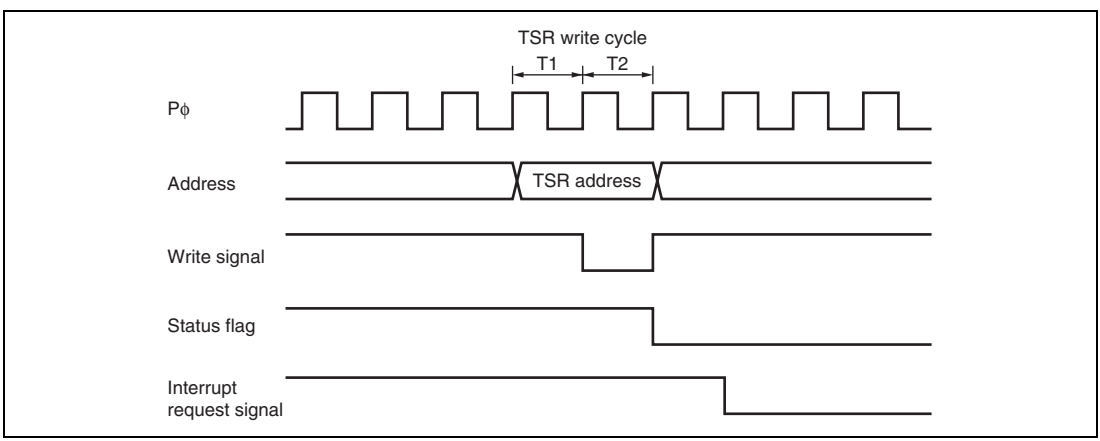


Figure 10.115 Timing for Status Flag Clearing by CPU (Channels 0 to 4)

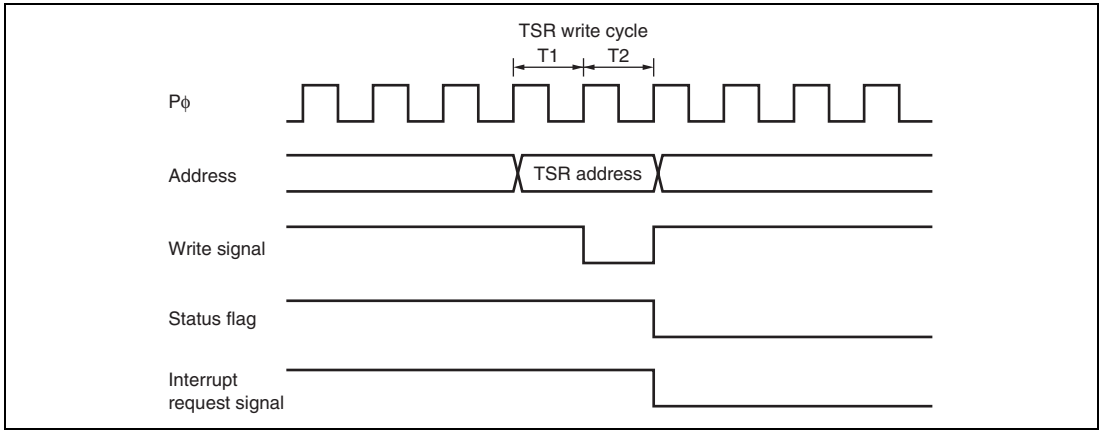


Figure 10.116 Timing for Status Flag Clearing by CPU (Channel 5)

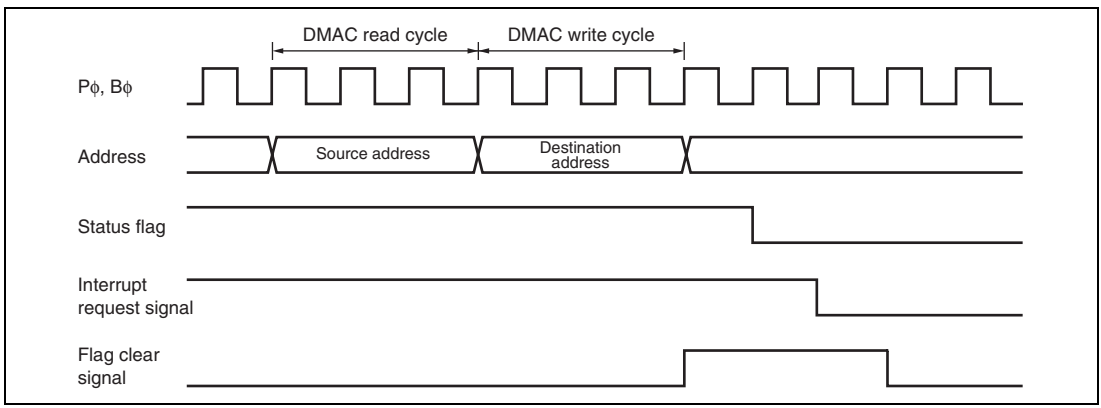


Figure 10.117 Timing for Status Flag Clearing by DTC Activation (Channels 0 to 4)

10.7 Usage Notes

10.7.1 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 23, Power-Down Modes.

10.7.2 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 10.118 shows the input clock conditions in phase counting mode.

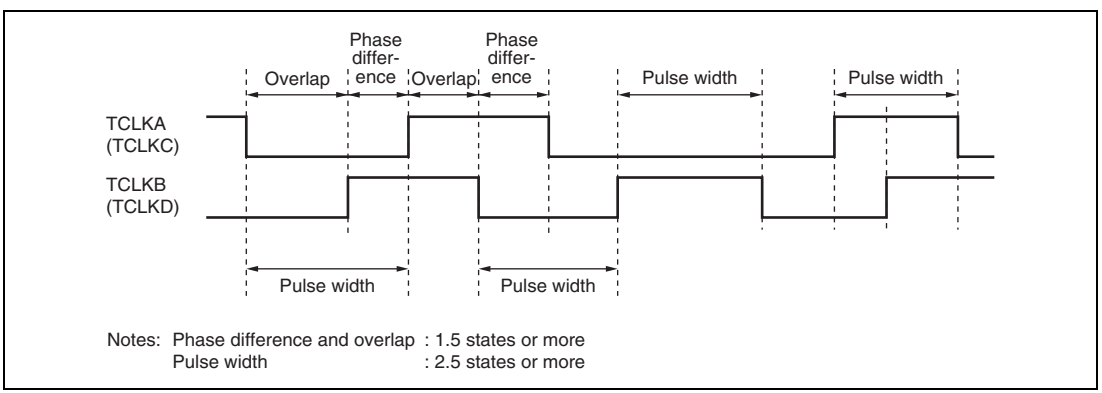


Figure 10.118 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

10.7.3 Caution on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

- Channel 0 to 4

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

- Channel 5

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

Where f: Counter frequency
Pϕ: Peripheral clock operating frequency
N: TGR set value

10.7.4 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 10.119 shows the timing in this case.

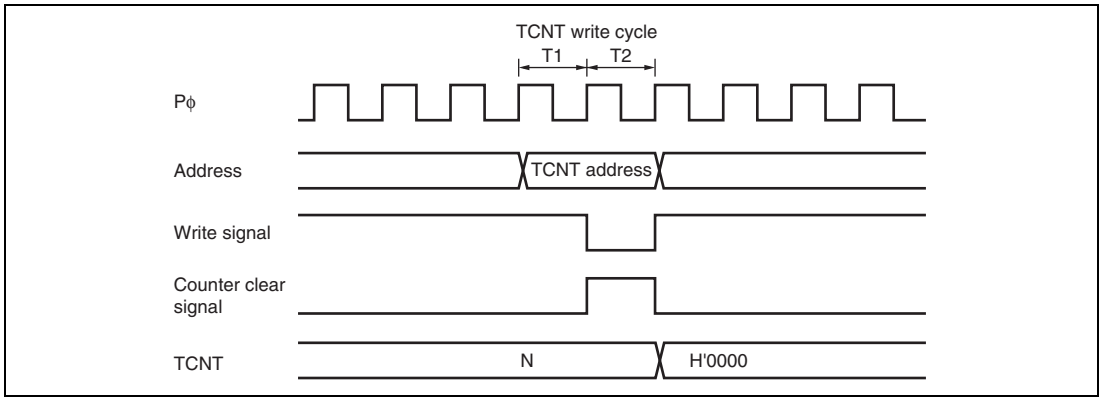


Figure 10.119 Contention between TCNT Write and Clear Operations

10.7.5 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 10.120 shows the timing in this case.

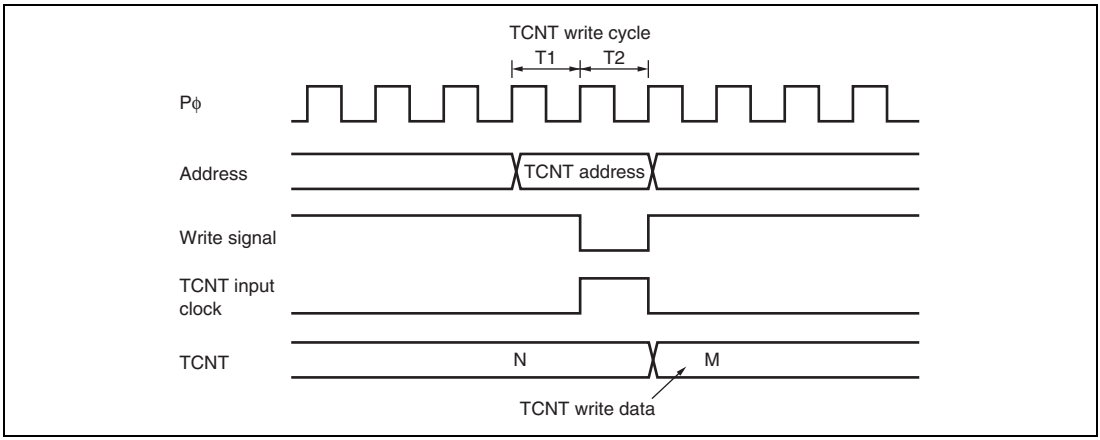


Figure 10.120 Contention between TCNT Write and Increment Operations

10.7.6 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 10.121 shows the timing in this case.

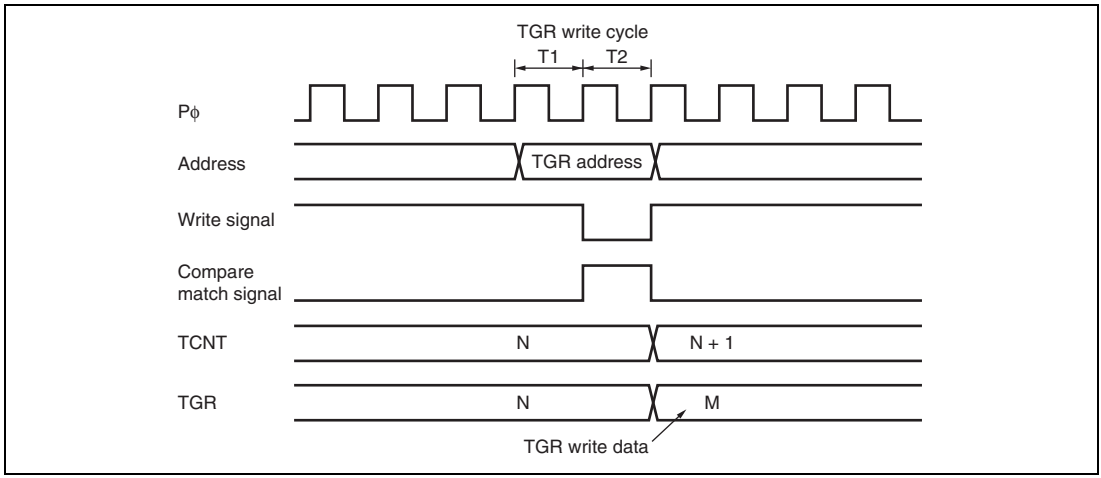


Figure 10.121 Contention between TGR Write and Compare Match

10.7.7 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 10.122 shows the timing in this case.

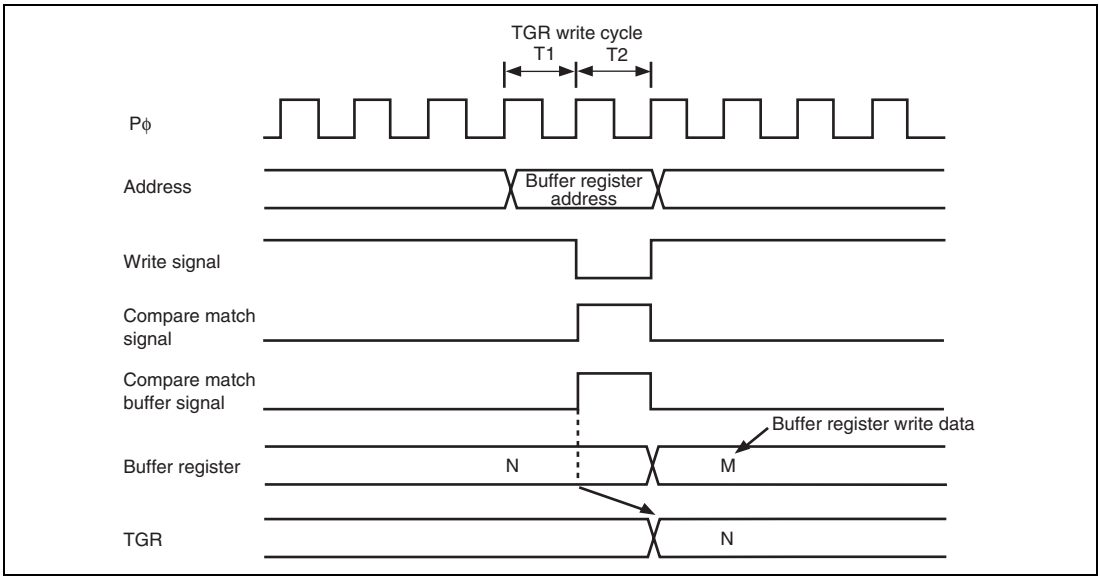


Figure 10.122 Contention between Buffer Register Write and Compare Match

10.7.8 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 10.123 shows the timing in this case.

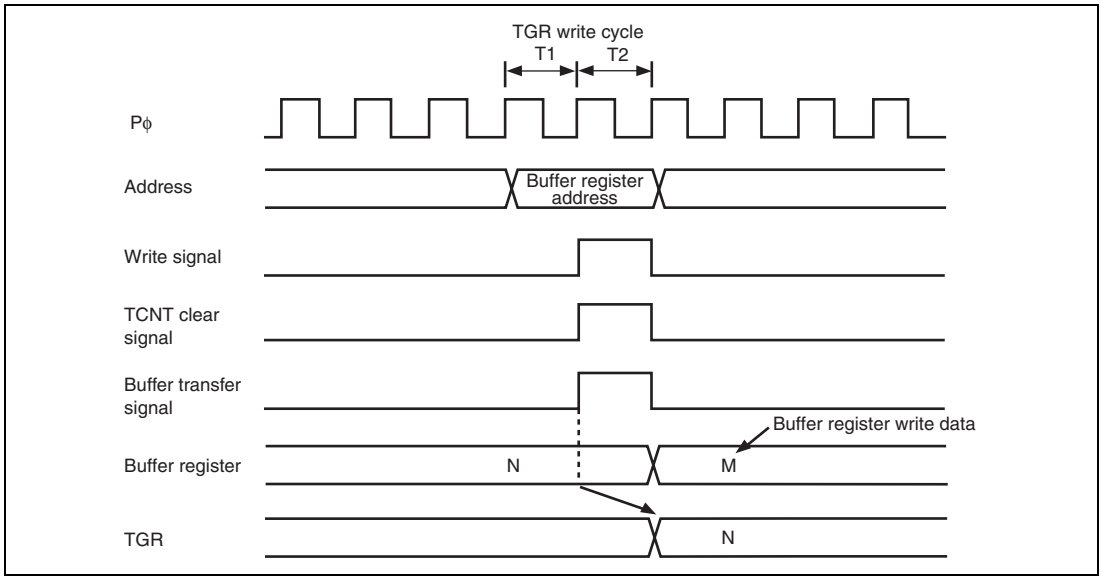


Figure 10.123 Contention between Buffer Register Write and TCNT Clear

10.7.9 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 10.124 and 125 show the timing in this case.

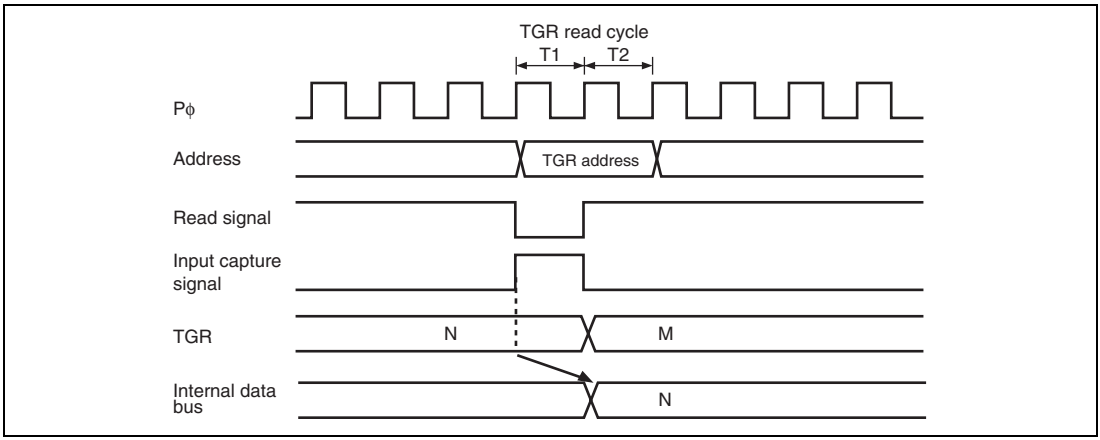


Figure 10.124 Contention between TGR Read and Input Capture (Channels 0 to 4)

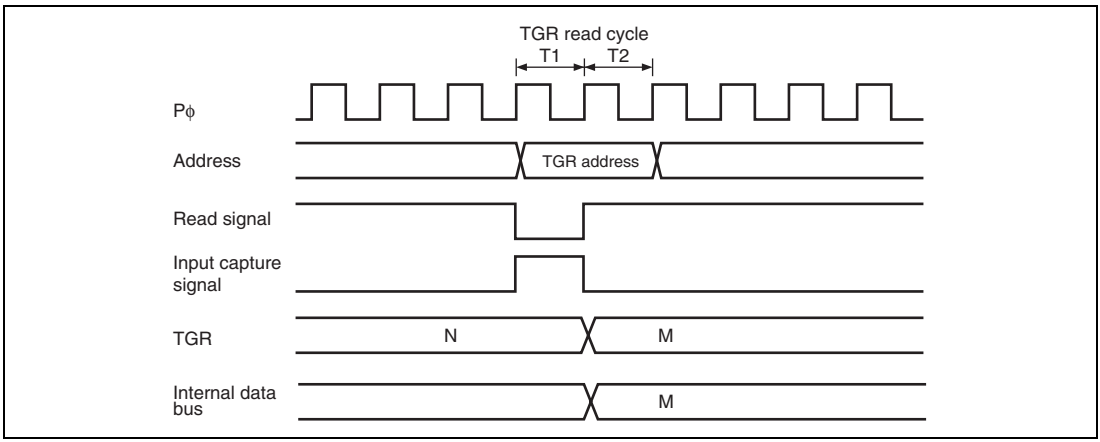


Figure 10.125 Contention between TGR Read and Input Capture (Channel 5)

10.7.10 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 10.126 and 127 show the timing in this case.

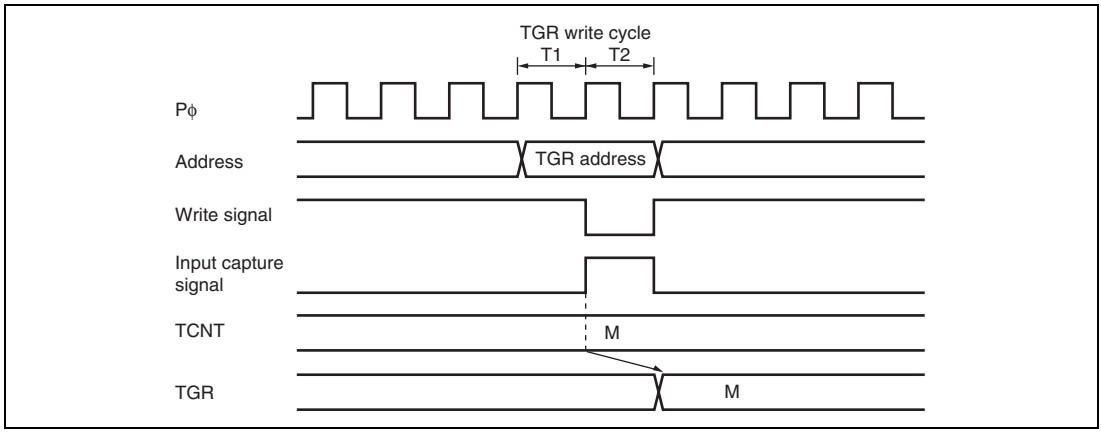


Figure 10.126 Contention between TGR Write and Input Capture (Channels 0 to 4)

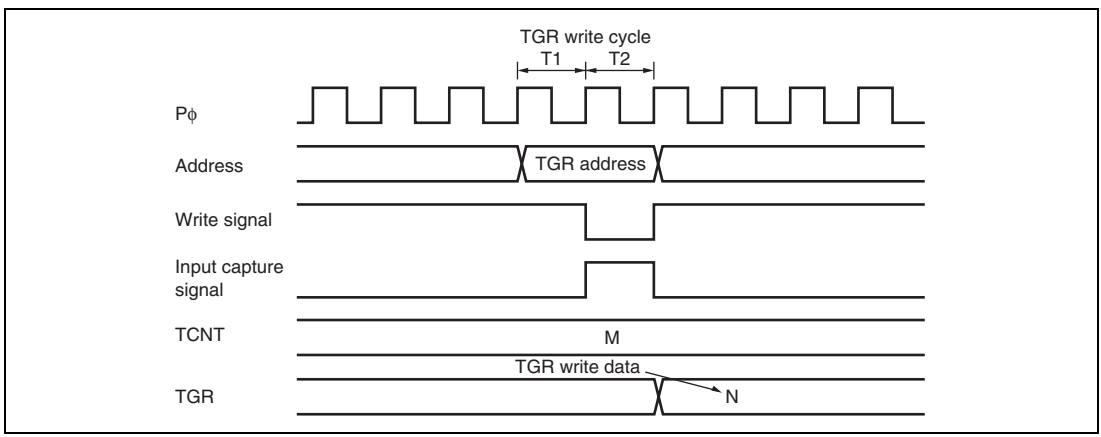


Figure 10.127 Contention between TGR Write and Input Capture (Channel 5)

10.7.11 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 10.128 shows the timing in this case.

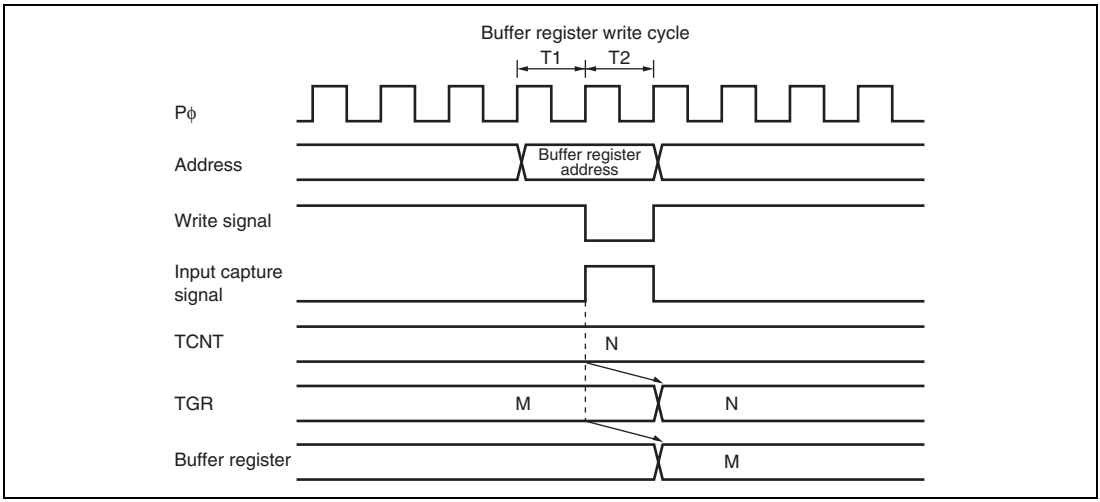


Figure 10.128 Contention between Buffer Register Write and Input Capture

10.7.12 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 10.129.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

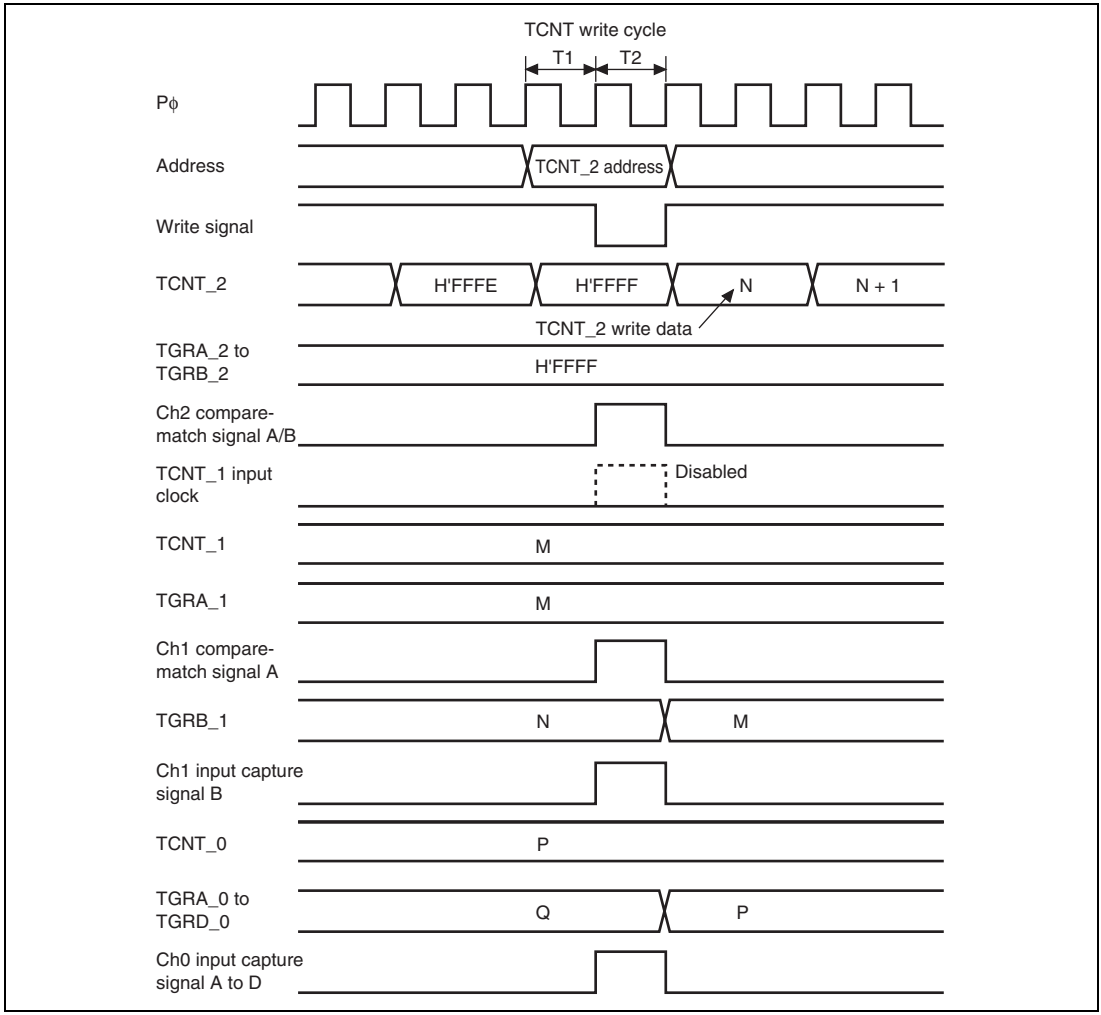


Figure 10.129 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

10.7.13 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 10.130.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

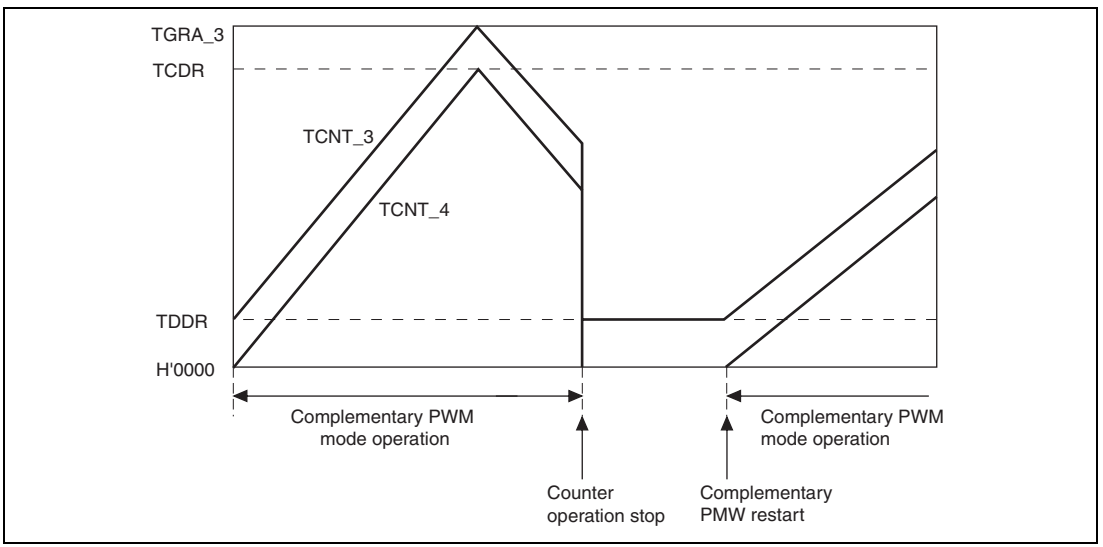


Figure 10.130 Counter Value during Complementary PWM Mode Stop

10.7.14 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.

10.7.15 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 10.131 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.

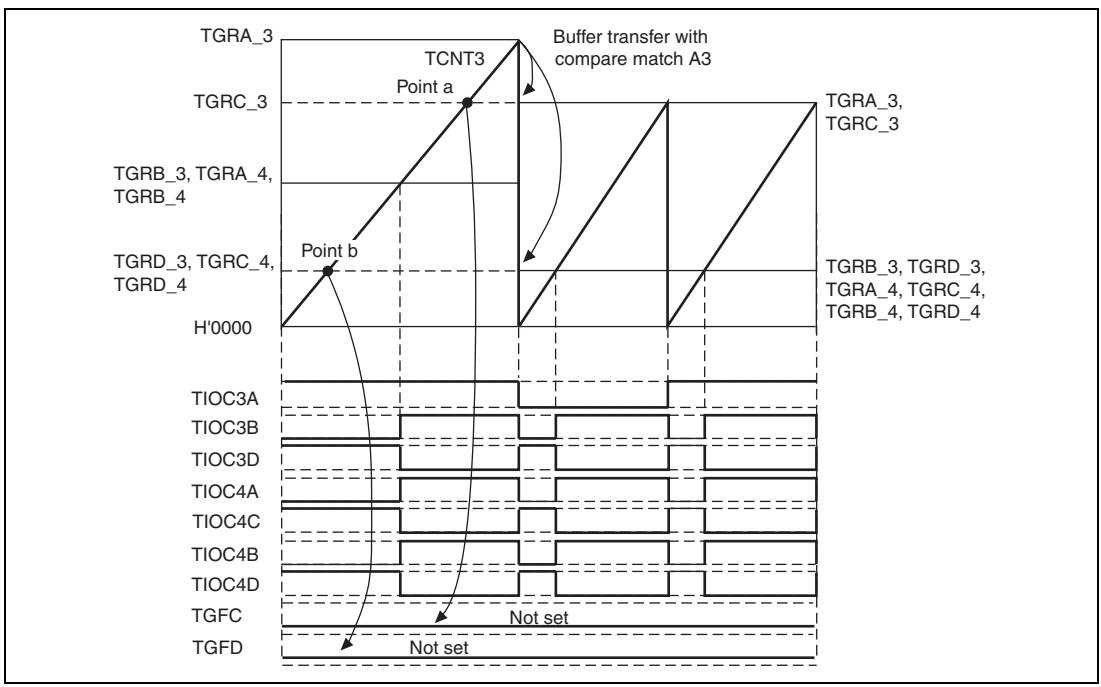


Figure 10.131 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

10.7.16 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 10.132 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.

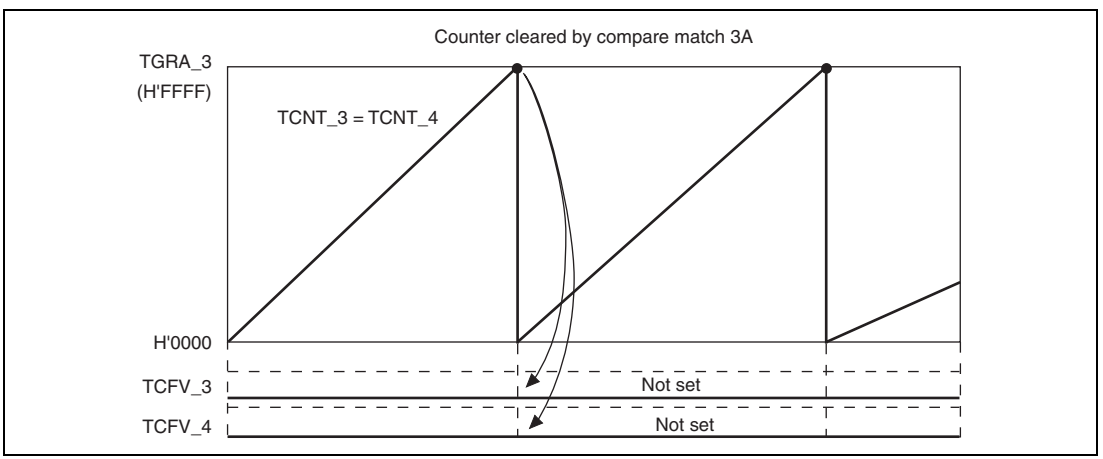


Figure 10.132 Reset Synchronous PWM Mode Overflow Flag

10.7.17 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 10.133 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

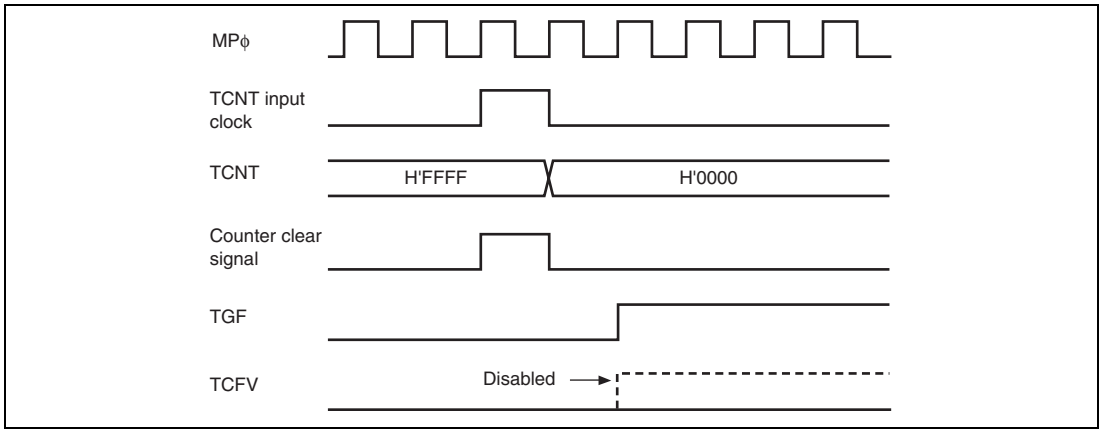


Figure 10.133 Contention between Overflow and Counter Clearing

10.7.18 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 10.134 shows the operation timing when there is contention between TCNT write and overflow.

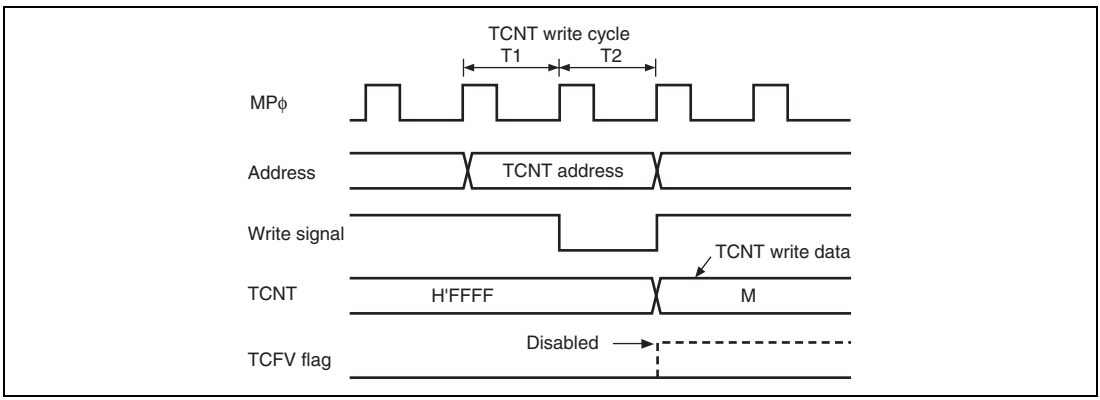


Figure 10.134 Contention between TCNT Write and Overflow

10.7.19 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.

10.7.20 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.

10.7.21 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.

10.7.22 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 additionally supports the function that can capture TCNT_1 and TCNT_2 simultaneously via a single input capture input. This function allows 32-bit counter fetches without TCNT_1 and TCNT_2 capture timing deviation. For details, see section 10.3.8, Timer Input Capture Control Register (TICCR).

10.8 MTU2 Output Pin Initialization

10.8.1 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.

10.8.2 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 *.

10.8.3 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 10.59.

Table 10.59 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

10.8.4 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 10.59. 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 10.135 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.

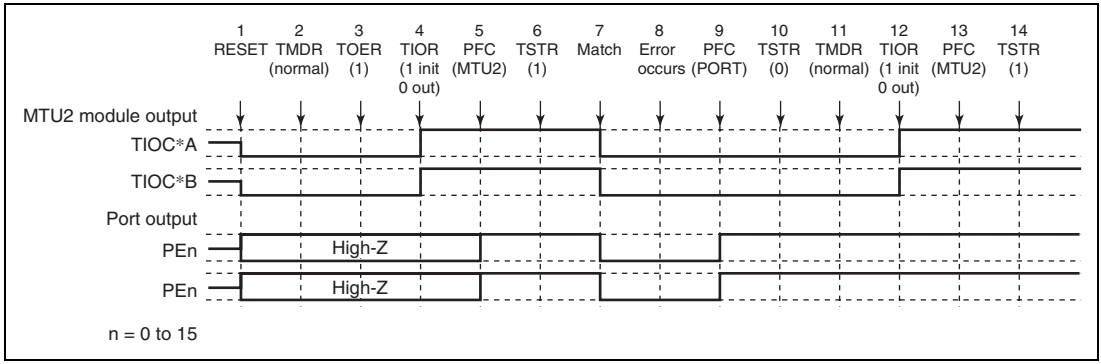


Figure 10.135 Error Occurrence in Normal Mode, Recovery in Normal Mode

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 10.136 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.

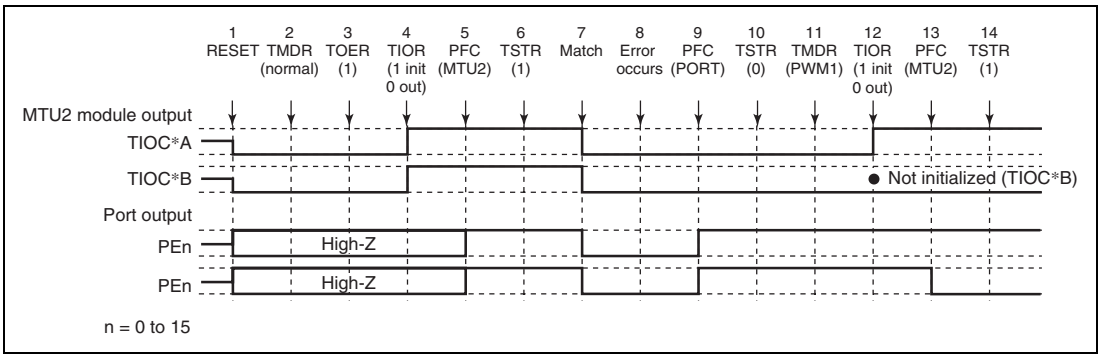


Figure 10.136 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 10.135.

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 10.137 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.

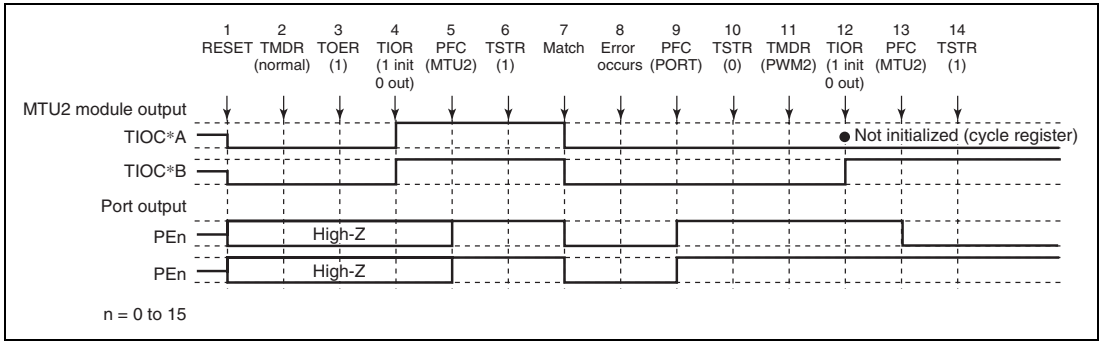


Figure 10.137 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 10.135.

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 10.138 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.

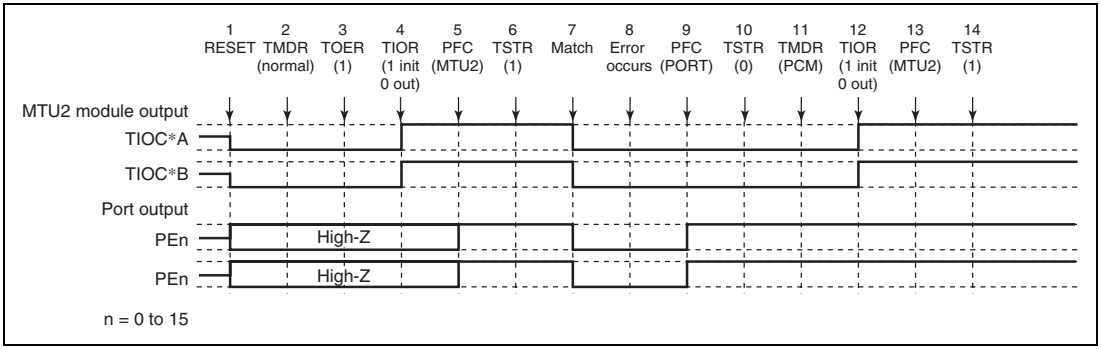


Figure 10.138 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 10.135.

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 10.139 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.

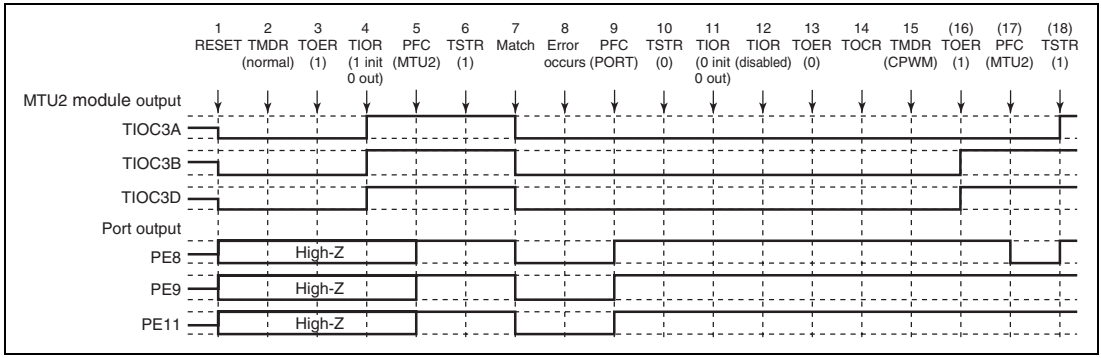


Figure 10.139 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.135.

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 10.140 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.

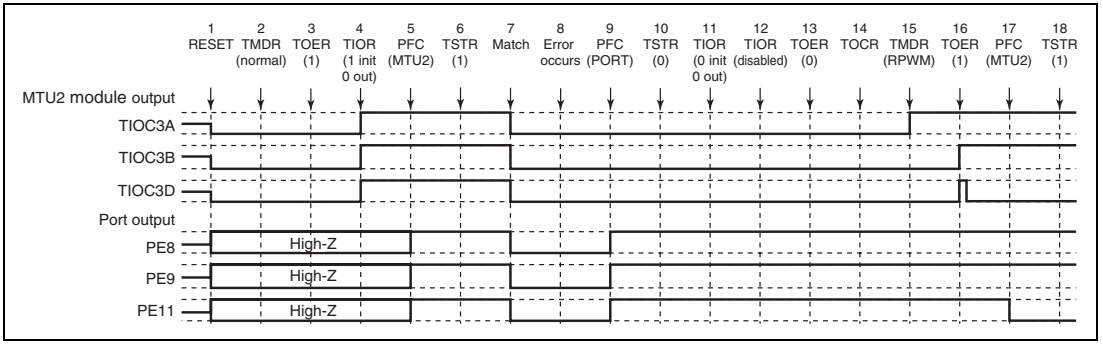


Figure 10.140 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 10.135.

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 10.141 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.

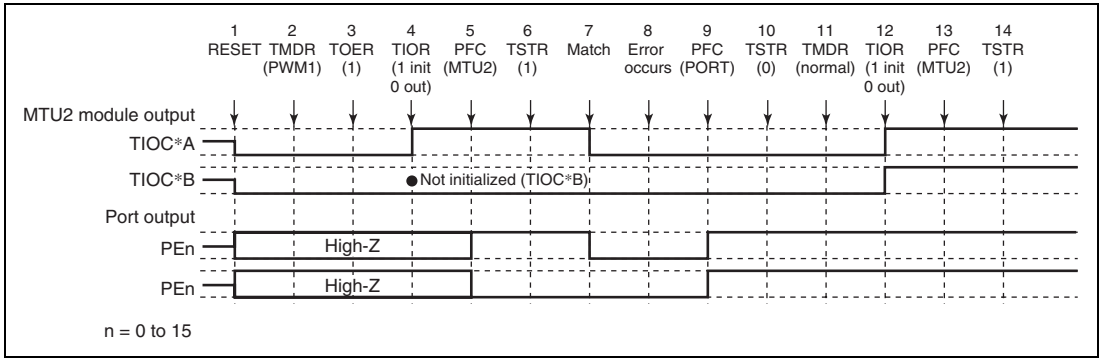


Figure 10.141 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

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 10.142 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.

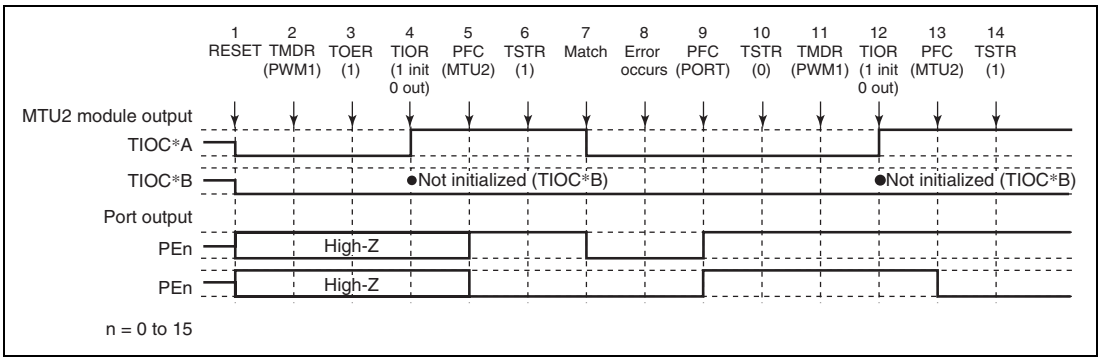


Figure 10.142 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 10.141.

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 10.143 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.

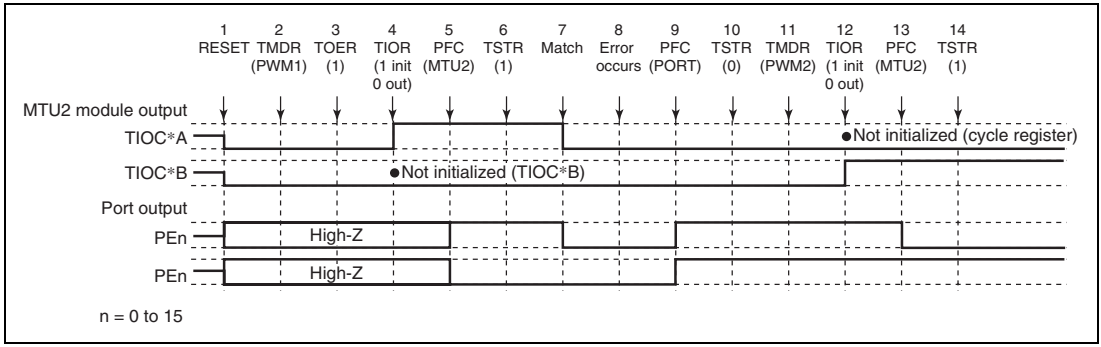


Figure 10.143 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

1 to 10 are the same as in figure 10.141.

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 10.144 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.

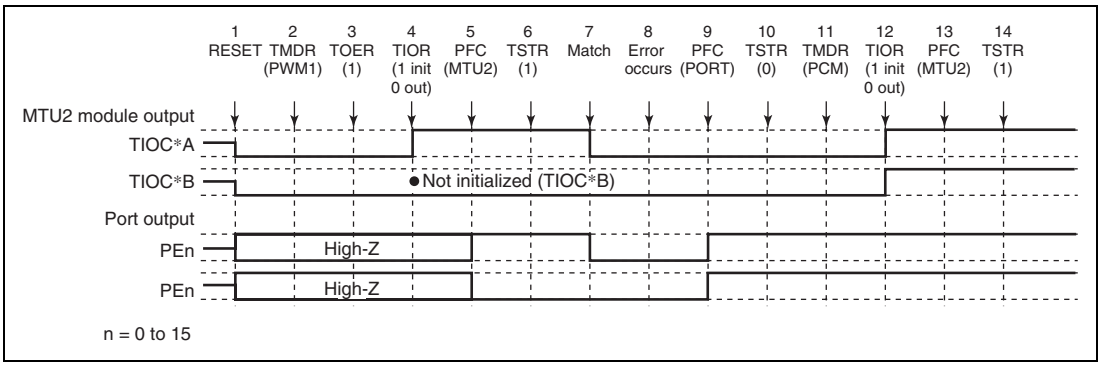


Figure 10.144 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 10.141.

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 10.145 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.

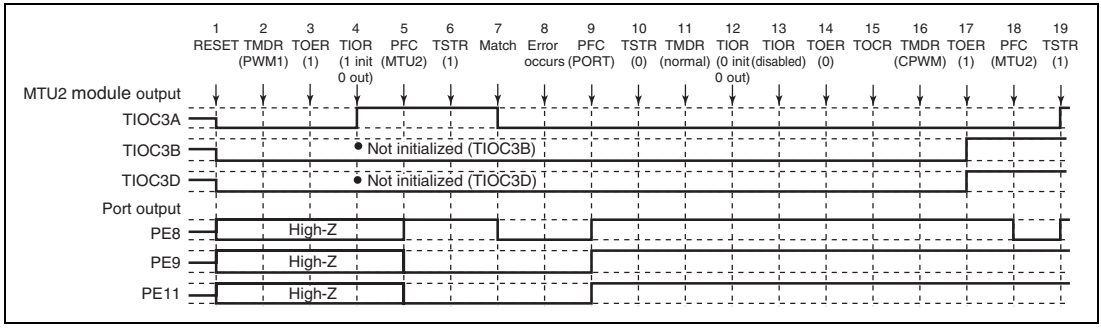


Figure 10.145 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.141.

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 10.146 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.

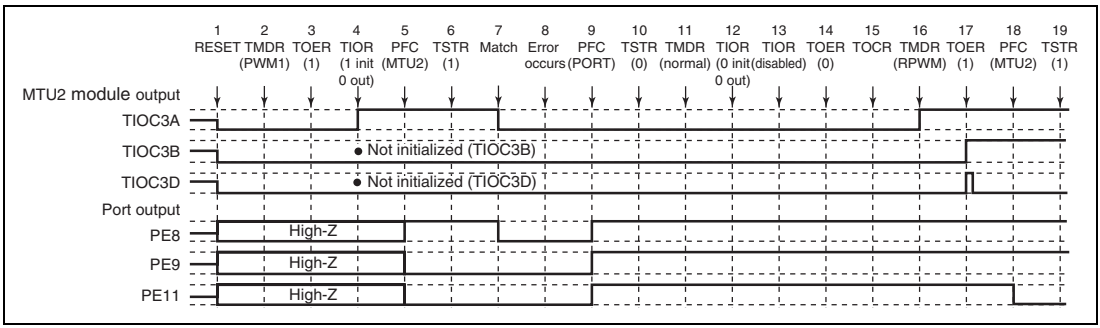


Figure 10.146 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 10.145.

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 10.147 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.

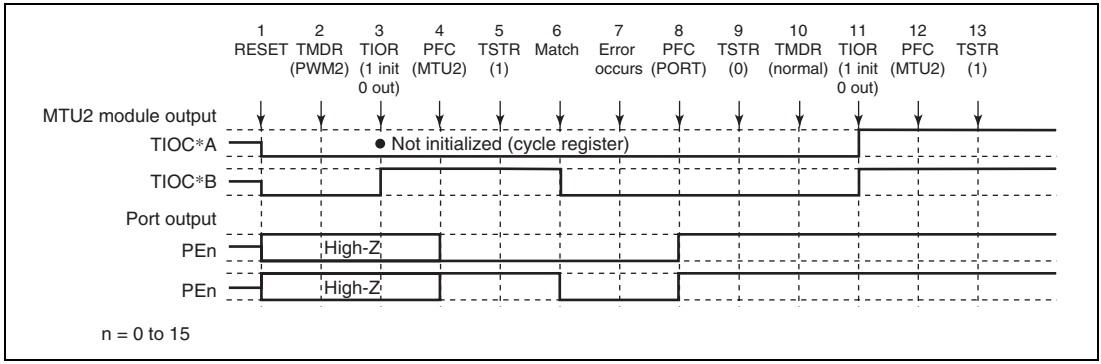


Figure 10.147 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

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 10.148 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.

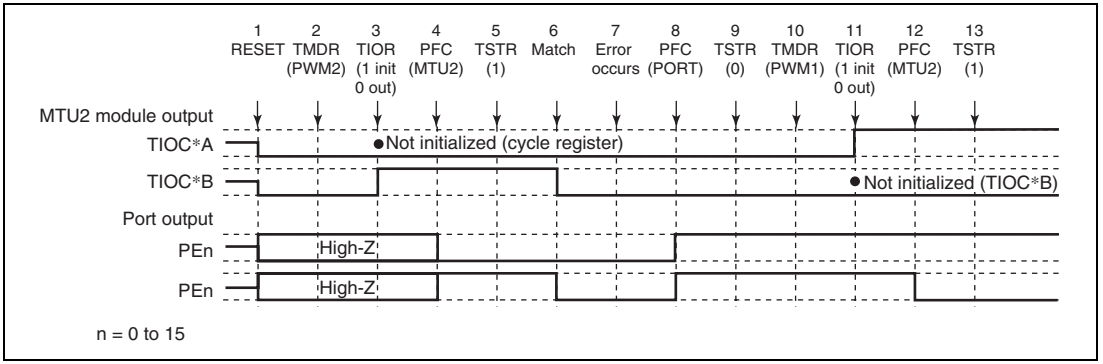


Figure 10.148 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 10.147.

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 10.149 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.

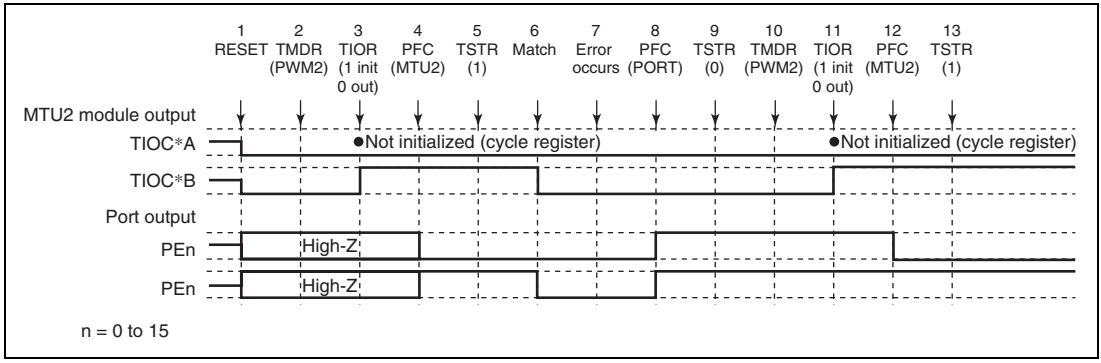


Figure 10.149 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 10.147.

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 10.150 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.

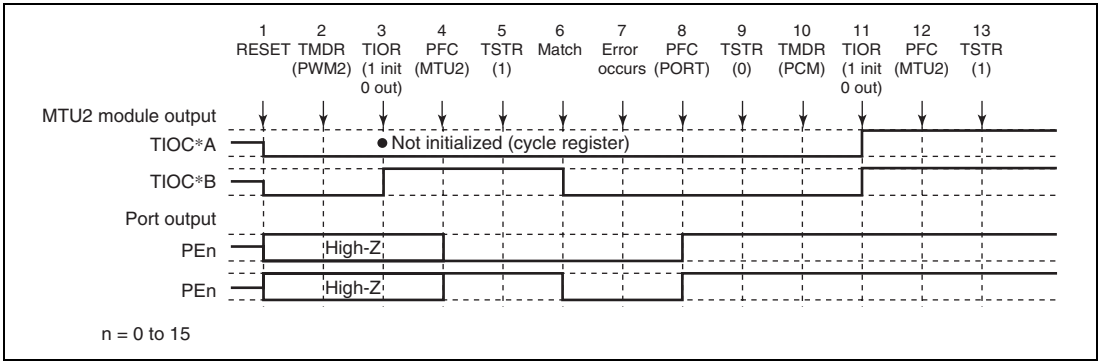


Figure 10.150 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 10.147.

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 10.151 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.

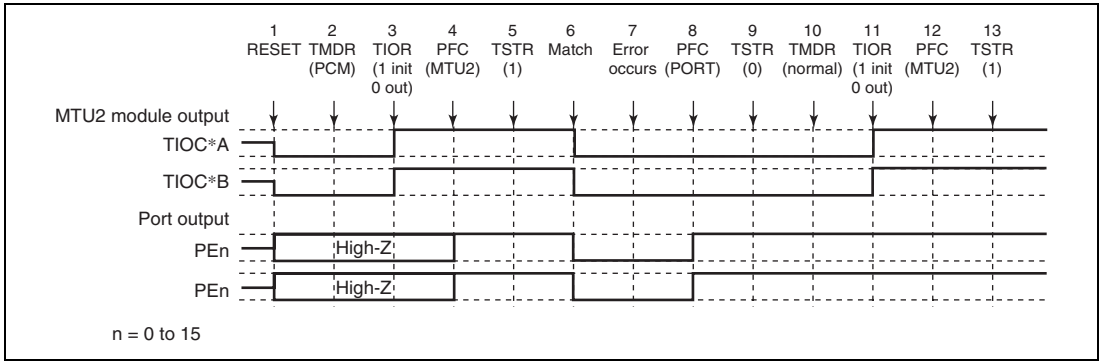


Figure 10.151 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

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 10.152 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.

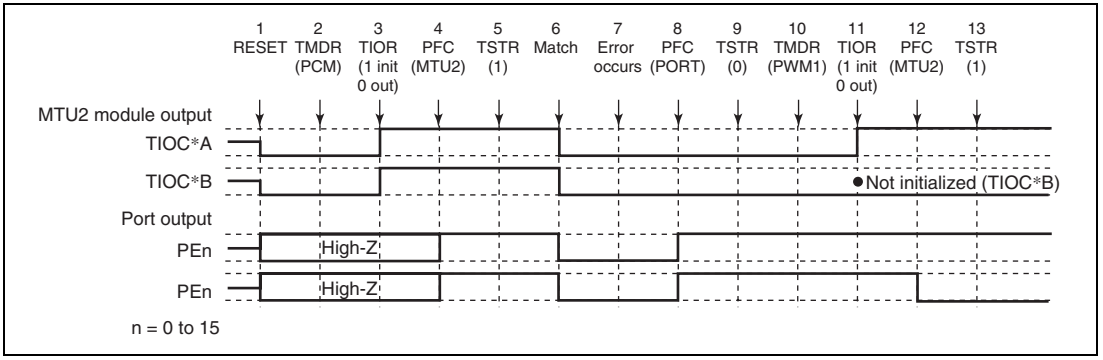


Figure 10.152 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 10.151.

10. Set PWM mode 1.
11. Initialize the pins with TIOR. (In PWM mode 1, the TIOC*B side is not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

(19) Operation when Error Occurs during Phase Counting Mode Operation, and Operation is Restarted in PWM Mode 2

Figure 10.153 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.

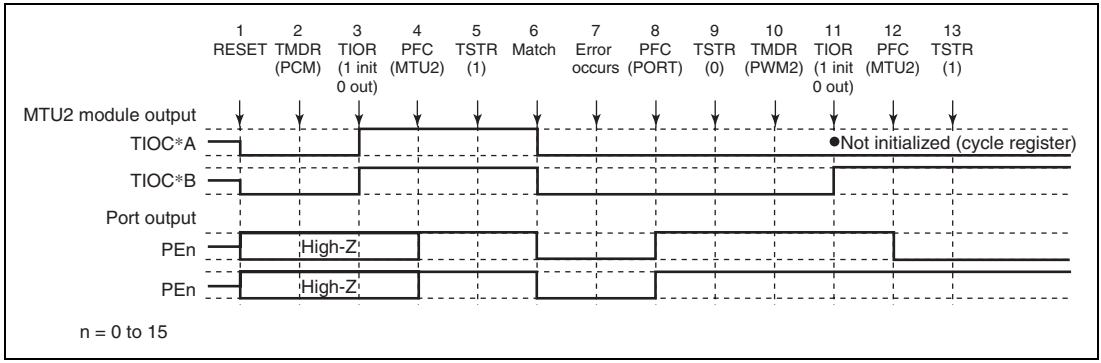


Figure 10.153 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 10.151.

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 10.154 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.

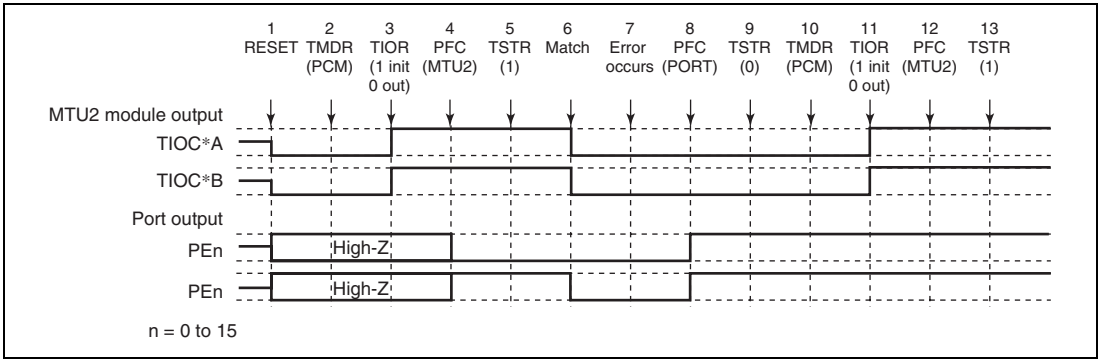


Figure 10.154 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 10.151.

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 10.155 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.

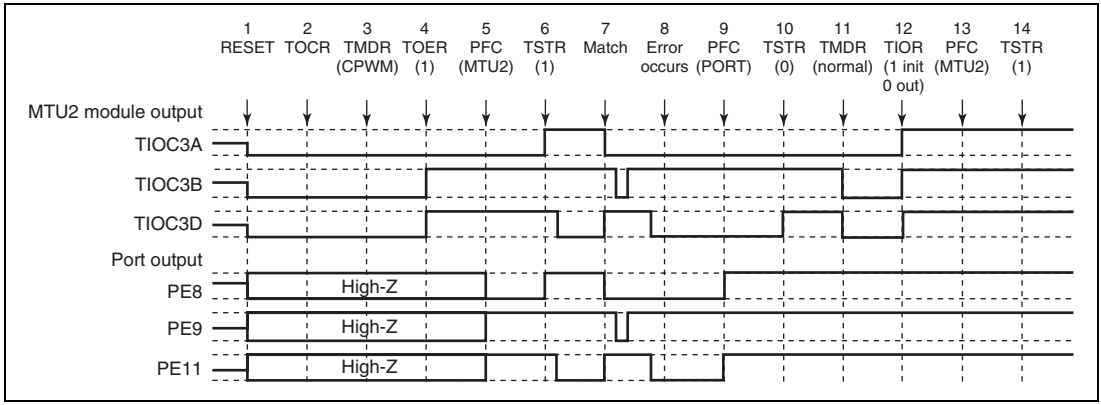


Figure 10.155 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

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 10.156 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.

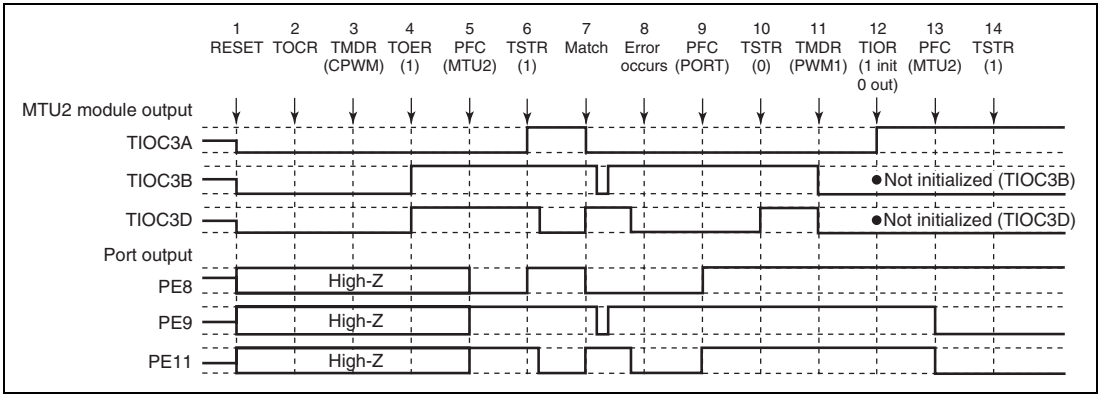


Figure 10.156 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 10.155.

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 10.157 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).

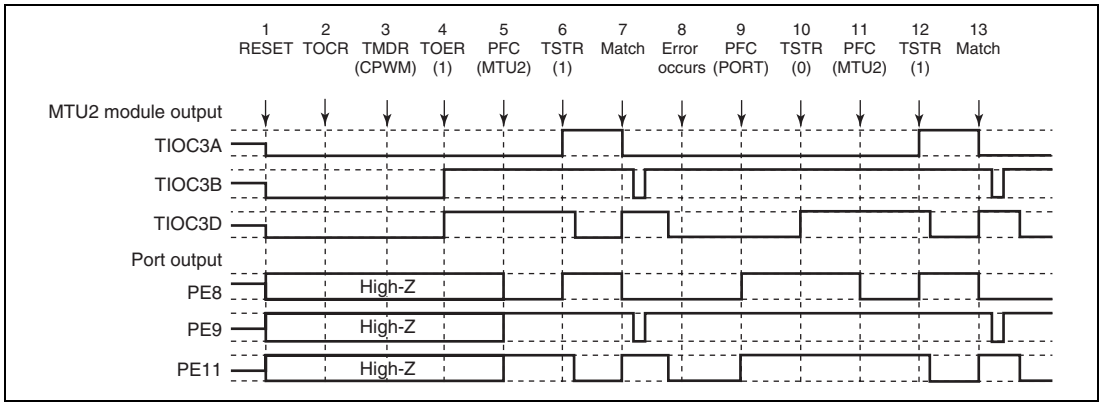


Figure 10.157 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.155.

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 10.158 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).

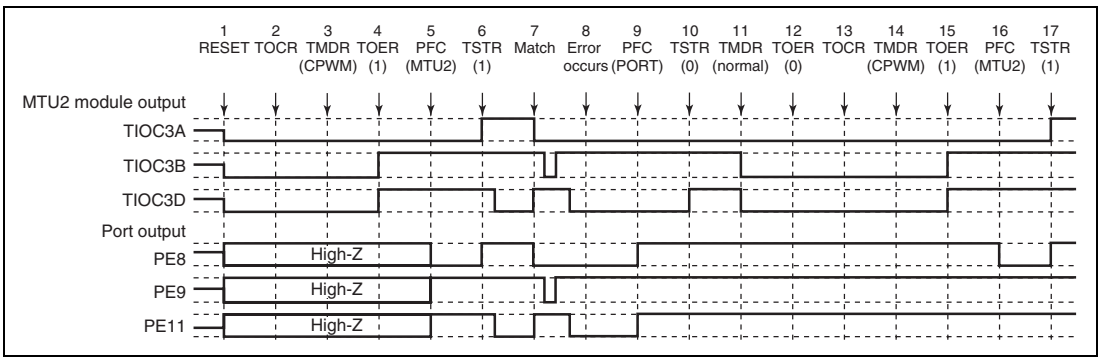


Figure 10.158 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.155.

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

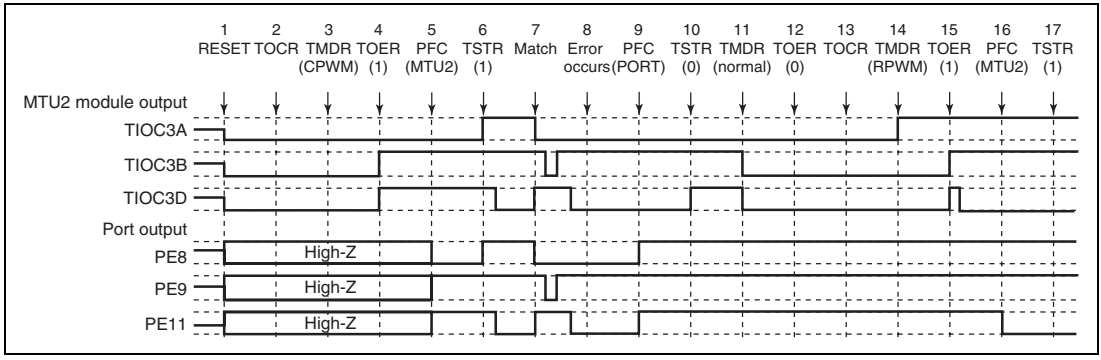


Figure 10.159 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 10.155.

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 10.160 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.

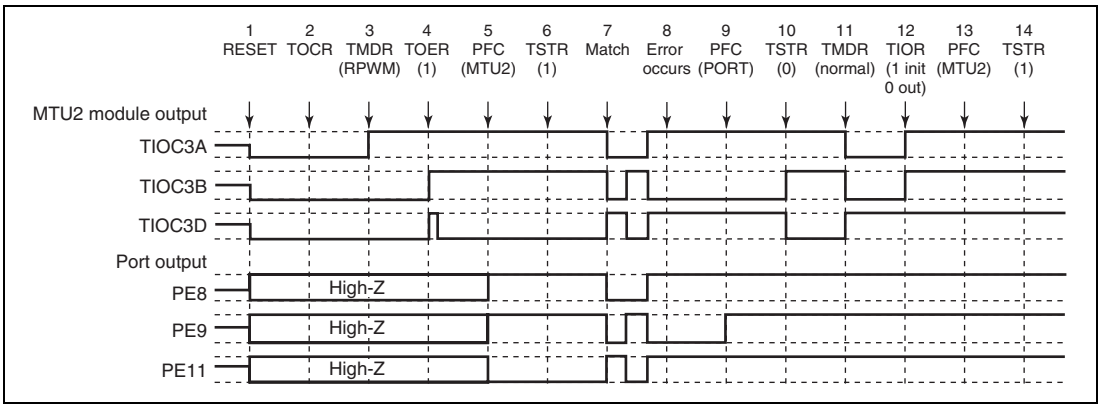


Figure 10.160 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

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 10.161 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.

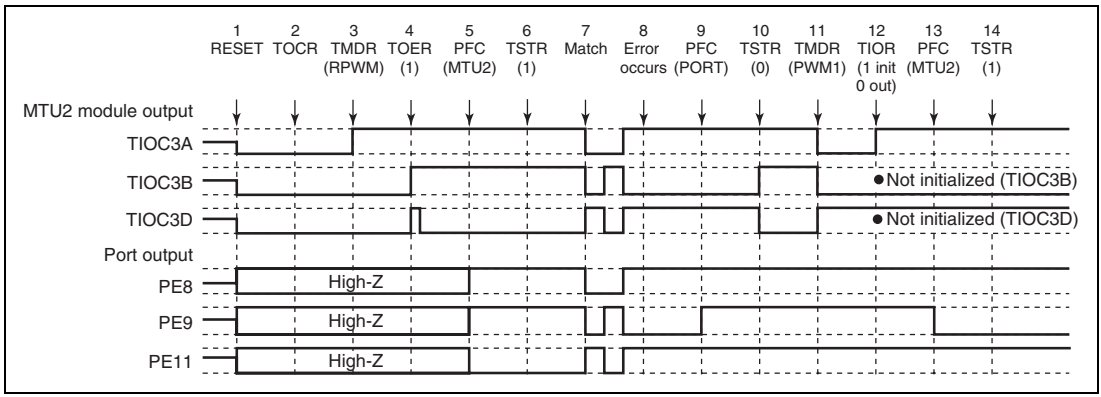


Figure 10.161 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 10.160.

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 10.162 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.

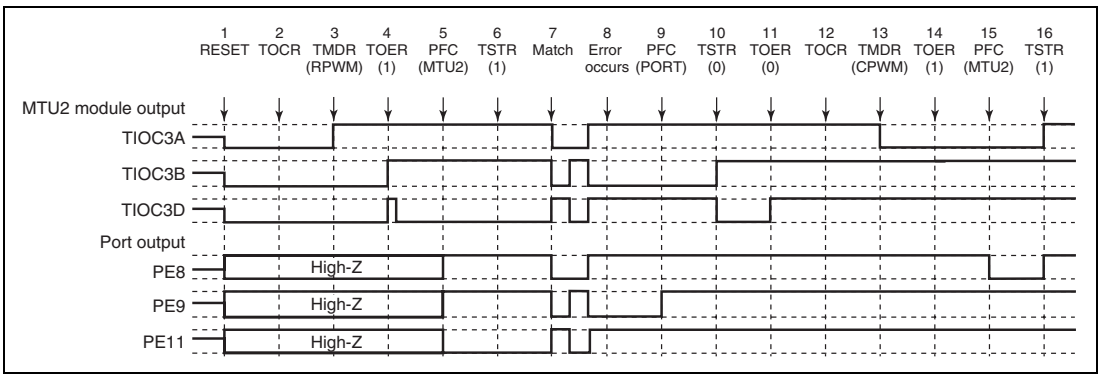


Figure 10.162 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 10.160.

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 10.163 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.

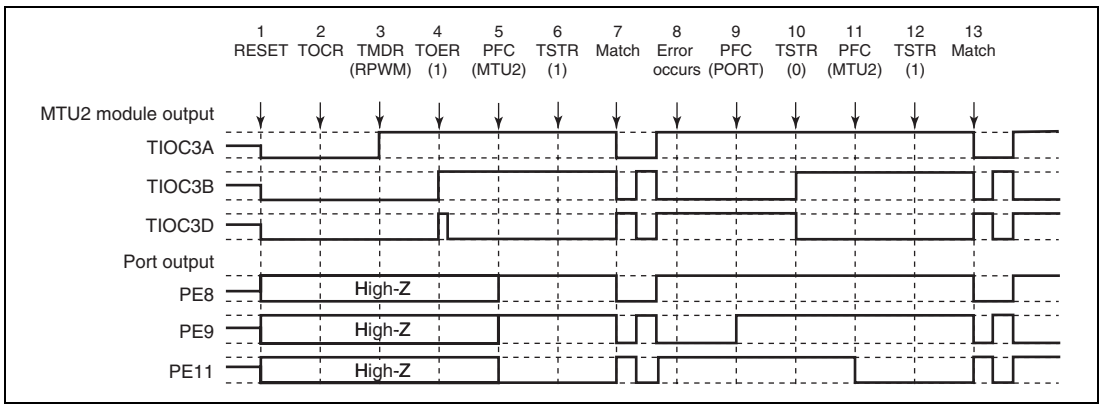


Figure 10.163 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 10.160.

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 11 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 10, Multi-Function Timer Pulse Unit 2 (MTU2). The MTU2S operates on Mϕ clock (MTU clock) while the MTU2 operates on Pϕ clock (peripheral clock) thus the term Pϕ in the MTU2 corresponds to Mϕ in the MTU2S. To distinguish from the MTU2, "S" is added to the end of the MTU2S input/output pin and register names. For example, TIOC3A is called TIOC3AS and TGRA_3 is called TGRA_3S in this section.

The MTU2S can operate at 80 MHz max. for complementary PWM output functions or at 40 MHz max. for the other functions.

Table 11.1 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		√	√	—

Item	Channel 3	Channel 4	Channel 5
Counter function of compensation for dead time	—	—	√
DMAC activation	—	—	—
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	—

Item	Channel 3	Channel 4	Channel 5
Interrupt skipping function	• Skips TGRA_3S compare match interrupts	• Skips TCIV_4S interrupts	—

[Legend]
√: Possible
—: Not possible

11.1 Input/Output Pins

Table 11.2 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

Note: For the pin configuration in complementary PWM mode, see table 10.54.

11.2 Register Descriptions

The MTU2S has the following registers. For details on register addresses and register states during each process, refer to section 26, 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 11.3 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
3	Timer control register_3S	TCR_3S	R/W	H'00	H'FFFE4A00	8
	Timer mode register_3S	TMDR_3S	R/W	H'00	H'FFFE4A02	8
	Timer I/O control register H_3S	TIORH_3S	R/W	H'00	H'FFFE4A04	8
	Timer I/O control register L_3S	TIORL_3S	R/W	H'00	H'FFFE4A05	8
	Timer interrupt enable register_3S	TIER_3S	R/W	H'00	H'FFFE4A08	8
	Timer status register_3S	TSR_3S	R/W	H'C0	H'FFFE4A2C	8
	Timer counter_3S	TCNT_3S	R/W	H'0000	H'FFFE4A10	16
	Timer general register A_3S	TGRA_3S	R/W	H'FFFF	H'FFFE4A18	16
	Timer general register B_3S	TGRB_3S	R/W	H'FFFF	H'FFFE4A1A	16
	Timer general register C_3S	TGRC_3S	R/W	H'FFFF	H'FFFE4A24	16
	Timer general register D_3S	TGRD_3S	R/W	H'FFFF	H'FFFE4A26	16
	Timer buffer operation transfer mode register_3S	TBTM_3S	R/W	H'00	H'FFFE4A38	8
4	Timer control register_4S	TCR_4S	R/W	H'00	H'FFFE4A01	8
	Timer mode register_4S	TMDR_4S	R/W	H'00	H'FFFE4A03	8
	Timer I/O control register H_4S	TIORH_4S	R/W	H'00	H'FFFE4A06	8
	Timer I/O control register L_4S	TIORL_4S	R/W	H'00	H'FFFE4A07	8
	Timer interrupt enable register_4S	TIER_4S	R/W	H'00	H'FFFE4A09	8
	Timer status register_4S	TSR_4S	R/W	H'C0	H'FFFE4A2D	8
	Timer counter_4S	TCNT_4S	R/W	H'0000	H'FFFE4A12	16
	Timer general register A_4S	TGRA_4S	R/W	H'FFFF	H'FFFE4A1C	16
	Timer general register B_4S	TGRB_4S	R/W	H'FFFF	H'FFFE4A1E	16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
4	Timer general register C_4S	TGRC_4S	R/W	H'FFFF	H'FFFE4A28	16
	Timer general register D_4S	TGRD_4S	R/W	H'FFFF	H'FFFE4A2A	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
	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
	Timer A/D converter start request cycle set buffer register B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFE4A4A	16
5	Timer control register U_5S	TCRU_5S	R/W	H'00	H'FFFE4884	8
	Timer control register V_5S	TCRV_5S	R/W	H'00	H'FFFE4894	8
	Timer control register W_5S	TCRW_5S	R/W	H'00	H'FFFE48A4	8
	Timer I/O control register U_5S	TIORU_5S	R/W	H'00	H'FFFE4886	8
	Timer I/O control register V_5S	TIORV_5S	R/W	H'00	H'FFFE4896	8
	Timer I/O control register W_5S	TIORW_5S	R/W	H'00	H'FFFE48A6	8
	Timer interrupt enable register_5S	TIER_5S	R/W	H'00	H'FFFE48B2	8
	Timer status register_5S	TSR_5S	R/W	H'00	H'FFFE48B0	8
	Timer start register_5S	TSTR_5S	R/W	H'00	H'FFFE48B4	8
	Timer counter U_5S	TCNTU_5S	R/W	H'0000	H'FFFE4880	16
	Timer counter V_5S	TCNTV_5S	R/W	H'0000	H'FFFE4890	16
	Timer counter W_5S	TCNTW_5S	R/W	H'0000	H'FFFE48A0	16
	Timer general register U_5S	TGRU_5S	R/W	H'FFFF	H'FFFE4882	16
	Timer general register V_5S	TGRV_5S	R/W	H'FFFF	H'FFFE4892	16
	Timer general register W_5S	TGRW_5S	R/W	H'FFFF	H'FFFE48A2	16
	Timer compare match clear register S	TCNTCMPCLRS	R/W	H'00	H'FFFE48B6	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Common	Timer start register S	TSTRS	R/W	H'00	H'FFFE4A80	8
	Timer synchronous register S	TSYRS	R/W	H'00	H'FFFE4A81	8
	Timer counter synchronous start register S	TCSYSTRS	R/W	H'00	H'FFFE4A82	8
	Timer read/write enable register S	TRWERS	R/W	H'01	H'FFFE4A84	8
Common to 3 and 4	Timer output master enable register S	TOERS	R/W	H'C0	H'FFFE4A0A	8
	Timer output control register 1S	TOCR1S	R/W	H'00	H'FFFE4A0E	8
	Timer output control register 2S	TOCR2S	R/W	H'00	H'FFFE4A0F	8
	Timer gate control register S	TGCRS	R/W	H80	H'FFFE4A0D	8
	Timer cycle control register S	TCDRS	R/W	H'FFFF	H'FFFE4A14	16
	Timer dead time data register S	TDDRS	R/W	H'FFFF	H'FFFE4A16	16
	Timer subcounter S	TCNTSS	R	H'0000	H'FFFE4A20	16
	Timer cycle buffer register S	TCBRS	R/W	H'FFFF	H'FFFE4A22	16
	Timer interrupt skipping set register S	TITCRS	R/W	H'00	H'FFFE4A30	8
	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 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 output level buffer register S	TOLBRS	R/W	H'00	H'FFFE4A36	8

Section 12 Port Output Enable 2 (POE2)

The port output enable 2 (POE2) can be used to place the high-current pins (PB18/TIOC3B, PB19/TIOC3D, PB4/TIOC4A, PB5/TIOC4B, PB6/TIOC4C, PB7/TIOC4D, PB12/TIOC4AS, PB13/TIOC4BS, PB10/TIOC4CS, PB11/TIOC4DS, PB21/TIOC3BS, and PB20/TIOC3DS) and the pins for channel 0 of the MTU2 (PA22/TIOC0A, PA23/TIOC0B, PA24/TIOC0C, and PA25/TIOC0D) in high-impedance state, depending on the change on the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, 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.

12.1 Features

- Each of the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, 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 $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, and $\overline{\text{POE8}}$ pin falling-edge or low-level sampling.
- High-current pins can be placed in high-impedance state when the high-current pin output levels are compared and simultaneous active-level output continues for one cycle or more.
- High-current pins and the pins for channel 0 of the MTU2 can be placed in high-impedance state by modifying the POE2 register settings.
- Interrupts can be generated by input-level sampling or output-level comparison results.

The POE2 has input level detection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in the block diagram of figure 12.1.

Figure 12.1 shows a block diagram of the POE2.

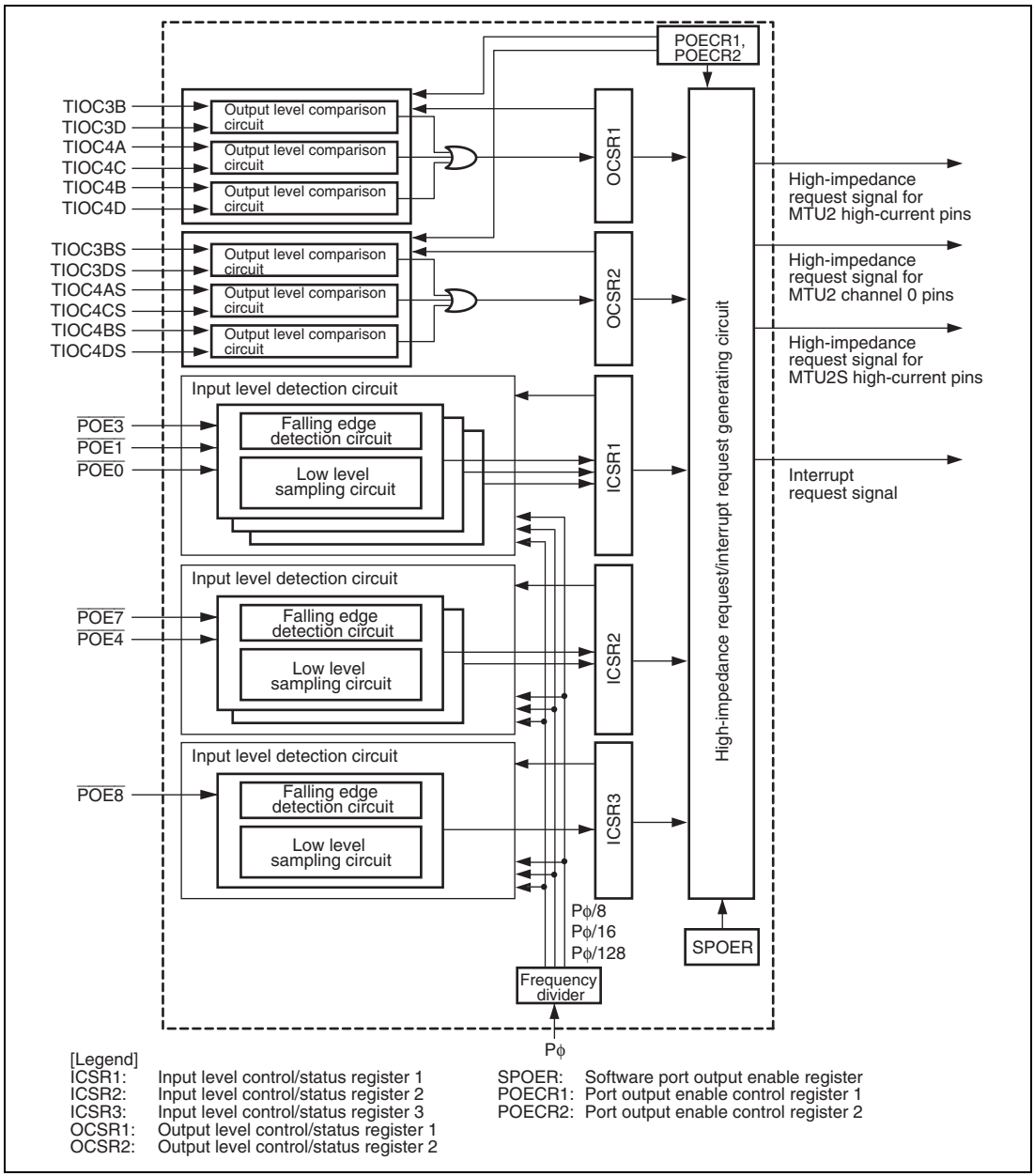


Figure 12.1 Block Diagram of POE2

12.2 Input/Output Pins

Table 12.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Port output enable input pins 0, 1, and 3	$\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$	Input	Input request signals to place high-current pins (PB18/TIOC3B, PB19/TIOC3D, PB4/TIOC4A, PB5/TIOC4B, PB6/TIOC4C, and PB7/TIOC4D) for MTU2 in high-impedance state
Port output enable input pins 4 and 7	$\overline{\text{POE4}}$ and $\overline{\text{POE7}}$	Input	Input request signals to place high-current pins (PB21/TIOC3BS, PB20/TIOC3DS, PB12/TIOC4AS, PB13/TIOC4BS, PB10/TIOC4CS, and PB11/TIOC4DS) for MTU2S in high-impedance state
Port output enable input pin 8	$\overline{\text{POE8}}$	Input	Inputs a request signal to place pins (PA22/TIOC0A, PA23/TIOC0B, PA24/TIOC0C, and PA25/TIOC0D) for channel 0 in MTU2 in high-impedance state

Table 12.2 shows output-level comparisons with pin combinations.

Table 12.2 Pin Combinations

Pin Combination	I/O	Description
PB18/TIOC3B and PB19/TIOC3D PB4/TIOC4A and PB6/TIOC4C PB5/TIOC4B and PB7/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.
PB21/TIOC3BS and PB20/TIOC3DS PB12/TIOC4AS and PB10/TIOC4CS PB13/TIOC4BS and PB11/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.

12.3 Register Descriptions

The POE2 has the following registers.

All these registers are initialized by a power-on reset, but are not initialized by a manual reset or in sleep mode, software standby mode, or module standby mode.

Table 12.3 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

12.3.1 Input Level Control/Status Register 1 (ICSR1)

ICSR1 is a 16-bit readable/writable register that selects the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, and $\overline{\text{POE3}}$ pin input modes, controls the enable/disable of interrupts, and indicates status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE3F	-	POE1F	POE0F	-	-	-	PIE1	POE3M[1:0]		-	-	POE1M[1:0]		POE0M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R	R	R/W*2	R/W*2	R/W*2	R/W*2

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

Bit	Bit Name	Initial Value	R/W	Description
15	POE3F	0	R/(W)*1	POE3 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE3}}$ pin. [Clearing conditions] • By writing 0 to POE3F after reading POE3F = 1 (when the falling edge is selected by bits 7 and 6 in ICSR1) • By writing 0 to POE3F after reading POE3F = 1 after a high level input to $\overline{\text{POE3}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 7 and 6 in ICSR1) [Setting condition] • When the input set by bits 7 and 6 in ICSR1 occurs at the $\overline{\text{POE3}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
14	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
13	POE1F	0	R/(W)*1	POE1 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE1}}$ pin. [Clearing conditions] • By writing 0 to POE1F after reading POE1F = 1 (when the falling edge is selected by bits 3 and 2 in ICSR1) • By writing 0 to POE1F after reading POE1F = 1 after a high level input to $\overline{\text{POE1}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 3 and 2 in ICSR1) [Setting condition] • When the input set by bits 3 and 2 in ICSR1 occurs at the $\overline{\text{POE1}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
12	POE0F	0	R/(W)*[1]	POE0 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE0}}$ pin. [Clear conditions] • By writing 0 to POE0F after reading POE0F = 1 (when the falling edge is selected by bits 1 and 0 in ICSR1) • By writing 0 to POE0F after reading POE0F = 1 after a high level input to $\overline{\text{POE0}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 1 and 0 in ICSR1) [Set condition] • When the input set by bits 1 and 0 in ICSR1 occurs at the $\overline{\text{POE0}}$ pin
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PIE1	0	R/W	Port Interrupt Enable 1 Enables or disables interrupt requests when any one of the POE0F, POE1F, and POE3F bits of the ICSR1 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE3M[1:0]	00	R/W*[2]	POE3 Mode These bits select the input mode of the $\overline{\text{POE3}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE3}}$ input 01: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE3}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3, 2	POE1M[1:0]	00	R/W*[2]	POE1 Mode These bits select the input mode of the $\overline{\text{POE1}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE1}}$ input 01: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE1}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.
1, 0	POE0M[1:0]	00	R/W*[2]	POE0 Mode These bits select the input mode of the $\overline{\text{POE0}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE0}}$ input 01: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are low level. 10: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are low level. 11: Accept request when $\overline{\text{POE0}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are low level.

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
2. Can be modified only once after a power-on reset.

12.3.2 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.

12.3.3 Input Level Control/Status Register 2 (ICSR2)

ICSR2 is a 16-bit readable/writable register that selects the $\overline{\text{POE4}}$ and $\overline{\text{POE7}}$ 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
	POE7F	-	-	POE4F	-	-	-	PIE2	POE7M[1:0]		-	-	-	-	POE4M[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	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	POE7F	0	R/(W)*1	POE7 Flag Indicates that a high impedance request has been input to the $\overline{\text{POE7}}$ pin. [Clearing conditions] • By writing 0 to POE7F after reading POE7F = 1 (when the falling edge is selected by bits 7 and 6 in ICSR2) • By writing 0 to POE7F after reading POE7F = 1 after a high level input to $\overline{\text{POE7}}$ is sampled at Pϕ/8, Pϕ/16, or Pϕ/128 clock (when low-level sampling is selected by bits 7 and 6 in ICSR2) [Setting condition] • When the input condition set by bits 7 and 6 in ICSR2 occurs at the $\overline{\text{POE7}}$ pin

Bit	Bit Name	Initial Value	R/W	Description
14, 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
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 any one of the POE4F and POE7F bits of the ICSR2 is set to 1. 0: Interrupt requests disabled 1: Interrupt requests enabled
7, 6	POE7M[1:0]	00	R/W*[2]	POE7 Mode These bits select the input mode of the $\overline{\text{POE7}}$ pin. 00: Accept request on falling edge of $\overline{\text{POE7}}$ input 01: Accept request when $\overline{\text{POE7}}$ input has been sampled for 16 Pϕ/8 clock pulses and all are at a low level. 10: Accept request when $\overline{\text{POE7}}$ input has been sampled for 16 Pϕ/16 clock pulses and all are at a low level. 11: Accept request when $\overline{\text{POE7}}$ input has been sampled for 16 Pϕ/128 clock pulses and all are at a low level.

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	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.

12.3.4 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

Bit	Bit Name	Initial Value	R/W	Description
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.

12.3.5 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.

Bit	Bit Name	Initial Value	R/W	Description
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.
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 POE8 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.

Bit	Bit Name	Initial Value	R/W	Description
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.

12.3.6 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.

Bit	Bit Name	Initial Value	R/W	Description
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
0	MTU2CH34HIZ	0	R/W	MTU2 Channel 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

12.3.7 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 PA25ZE	MTU2 PA24ZE	MTU2 PA23ZE	MTU2 PA22ZE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	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 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	MTU2PA25ZE	0	R/W*	MTU2PA25 High-Impedance Enable Specifies whether to place the PA25/TIOC0D pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
2	MTU2PA24ZE	0	R/W*	MTU2PA24 High-Impedance Enable Specifies whether to place the PA24/TIOC0C pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state
1	MTU2PA23ZE	0	R/W*	MTU2PA23 High-Impedance Enable Specifies whether to place the PA23/TIOC0B pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
0	MTU2PA22ZE	0	R/W*	MTU2PA22 High-Impedance Enable Specifies whether to place the PA22/TIOC0A pin for channel 0 in the MTU2 in high-impedance state when either POE8F or MTU2CH0HIZ bit is set to 1. 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state

12.3.8 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	-	-	-	-	-	-	-	-
Initial value:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R	R	R	R	R	R	R

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

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	MTU2P1CZE	1	R/W*	MTU2 Port 1 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2 high-current PB18/TIOC3B and PB19/TIOC3D pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state 1: Compares output levels and places the pins in high-impedance state

Bit	Bit Name	Initial Value	R/W	Description
13	MTU2P2CZE	1	R/W*	MTU2 Port 2 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2 high-current PB4/TIOC4A and PB6/TIOC4C pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state 1: Compares output levels and places the pins in high-impedance state
12	MTU2P3CZE	1	R/W*	MTU2 Port 3 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2 high-current PB5/TIOC4B and PB7/TIOC4D pins and to place them in high-impedance state when the OSF1 bit is set to 1 while the OCE1 bit is 1 or when any one of the POE0F, POE1F, POE3F, and MTU2CH34HIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state 1: Compares output levels and places the pins in high-impedance state
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10	MTU2SP1CZE	1	R/W*	MTU2S Port 1 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PB21/TIOC3BS and PB20/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 any one of the POE4F, POE7F, and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.

Bit	Bit Name	Initial Value	R/W	Description
9	MTU2SP2CZE	1	R/W*	MTU2S Port 2 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PB12/TIOC4AS and PB10/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 any one of the POE4F, POE7F, and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
8	MTU2SP3CZE	1	R/W*	MTU2S Port 3 Output Comparison/High-Impedance Enable Specifies whether to compare output levels for the MTU2S high-current PB13/TIOC4BS and PB11/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 any one of the POE4F, POE7F, and MTU2SHIZ bits is set to 1. 0: Does not compare output levels or place the pins in high-impedance state. 1: Compares output levels and places the pins in high-impedance state.
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

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

12.4 Operation

Table 12.4 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 12.4 Target Pins and Conditions for High-Impedance Control

Pins	Conditions	Detailed Conditions
MTU2 high-current pins (PB18/TIOC3B and PB19/TIOC3D)	Input level detection, output level comparison, or SPOER setting	MTU2P1CZE • ((POE3F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PB4/TIOC4A and PB6/TIOC4C)	Input level detection, output level comparison, or SPOER setting	MTU2P2CZE • ((POE3F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2 high-current pins (PB5/TIOC4B and PB7/TIOC4D)	Input level detection, output level comparison, or SPOER setting	MTU2P3CZE • ((POE3F+POE1F+POE0F) + (OSF1 • OCE1) + (MTU2CH34HIZ))
MTU2S high-current pins (PB21/TIOC3BS and PB20/TIOC3DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP1CZE • ((POE4F+POE7F) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PB12/TIOC4AS and PB10/TIOC4CS)	Input level detection, output level comparison, or SPOER setting	MTU2SP2CZE • ((POE4F+POE7F) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2S high-current pins (PB13/TIOC4BS and PB11/TIOC4DS)	Input level detection, output level comparison, or SPOER setting	MTU2SP3CZE • ((POE4F+POE7F) + (OSF2 • OCE2) + (MTU2SHIZ))
MTU2 channel 0 pins (PA22/TIOC0A, PA23/TIOC0B, PA24/TIOC0C, and PA25/TIOC0D)	Input level detection or SPOER setting	((POE8F • POE8E) + (MTU2CH0HIZ))

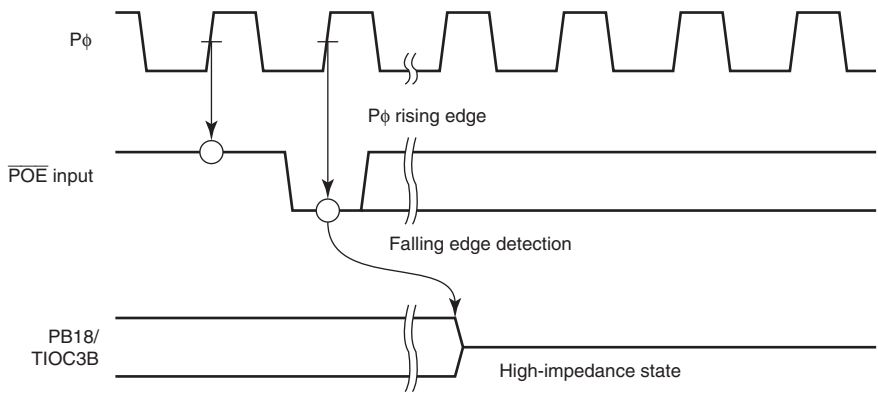
12.4.1 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR3 occur on the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, 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 $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, 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 12.2 shows the sample timing after the level changes in input to the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, 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 12.2 Falling Edge Detection

(2) Low-Level Detection

Figure 12.3 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.

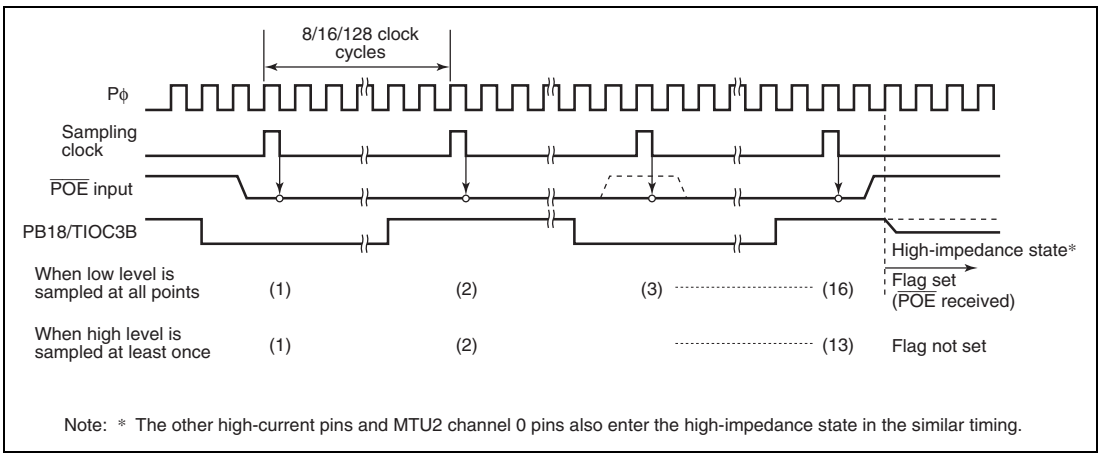


Figure 12.3 Low-Level Detection Operation

12.4.2 Output-Level Compare Operation

Figure 12.4 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.

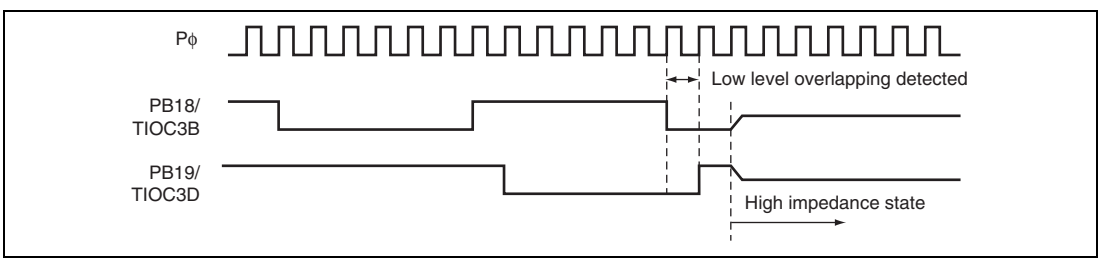


Figure 12.4 Output-Level Compare Operation

12.4.3 Release from High-Impedance State

High-current pins that have entered high-impedance state due to input-level detection can be released either by returning them to their initial state with a power-on reset, or by clearing all of the flags in bits 15 to 12 (POE8F, POE7F, POE4F, POE3F, POE1F, and POE0F) of ICSR1 to ICSR3. However, note that when low-level sampling is selected by bits 7 to 0 in ICSR1 to ICSR3, just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared by writing 0 to it only after a high level is input to one of the $\overline{\text{POE0}}$, $\overline{\text{POE1}}$, $\overline{\text{POE3}}$, $\overline{\text{POE4}}$, $\overline{\text{POE7}}$, 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.

12.5 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 12.5 shows the interrupt sources and their conditions.

Table 12.5 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE3F, POE1F, POE0F, and OSF1	PIE1 • (POE3F + POE1F + POE0F) + OIE1 • OSF1
OEI2	Output enable interrupt 2	POE8F	PIE3 • POE8F
OEI3	Output enable interrupt 3	POE4F, POE7F, and OSF2	PIE2 • (POE4F + POE7F) + OIE2 • OSF2

12.6 Usage Note

12.6.1 Pin Status When the WDT Issues a Power-On Reset

When a power-on reset is issued by the WDT, the pin function controller (PFC) is initialized and the I/O ports function as general inputs (initial value).

If a power-on reset is issued by the WDT during high-impedance processing by the port output enable (POE) signal, the I/O port pins are placed in output state for a time period of one cycle of the peripheral clock, Pϕ, until the pin functions switch to general inputs.

If a power-on reset is issued by the WDT during high-impedance processing by MTU2 or MTU2S short detection, the I/O port pins are placed in the same status as described above.

Figure 12.5 shows the I/O port pin status when a power-on reset is issued by the WDT during high-impedance processing by the POE input while the timer output is selected.

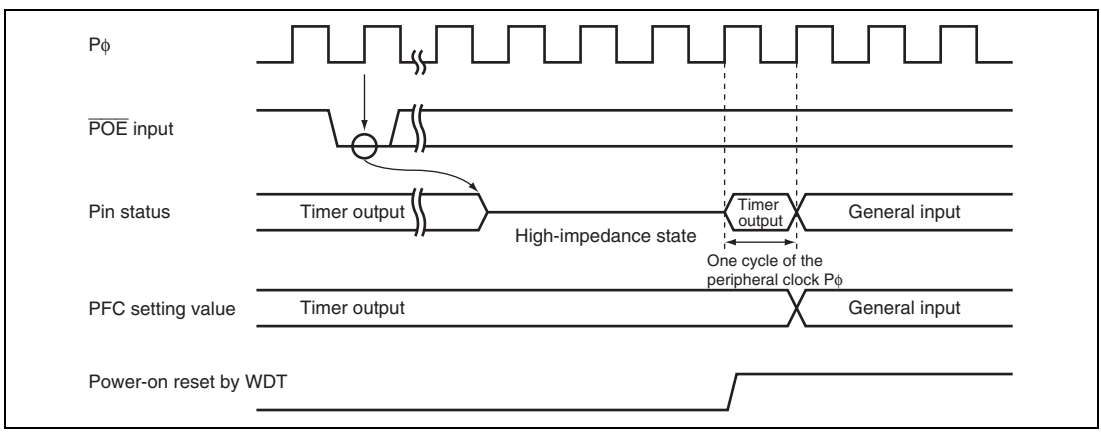


Figure 12.5 Pin Status When Power-on Reset is Issued from Watchdog Timer

Section 13 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.

13.1 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 DMA transfer request or interrupt request generation on compare match by DMAC setting
- When not in use, the CMT can be stopped by halting its clock supply to reduce power consumption.

Figure 13.1 shows a block diagram of CMT.

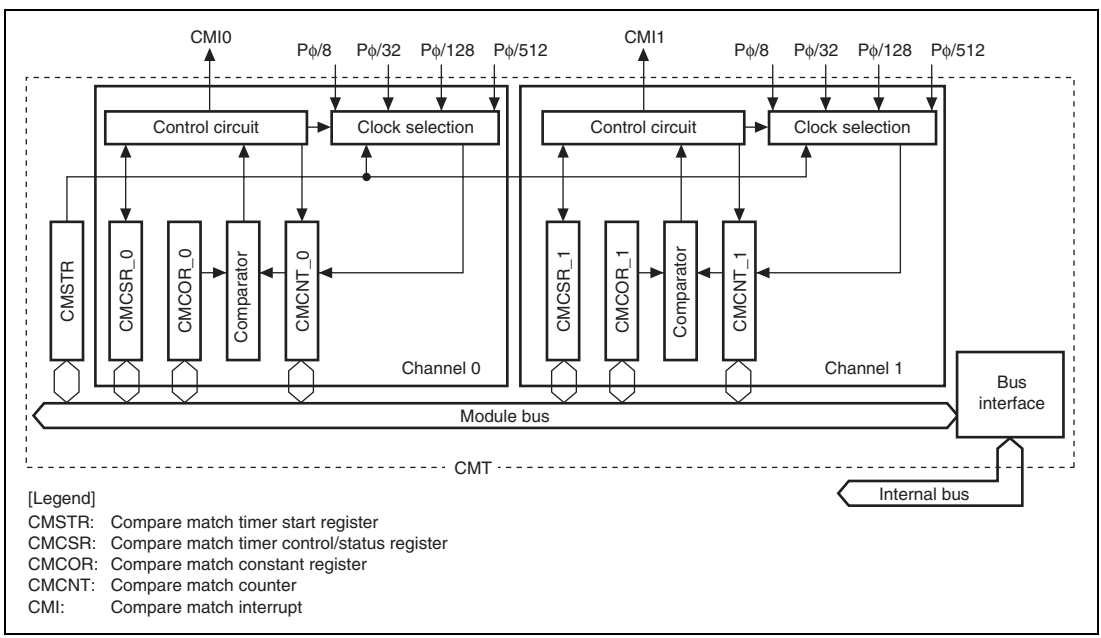


Figure 13.1 Block Diagram of CMT

13.2 Register Descriptions

The CMT has the following registers.

Table 13.1 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

13.2.1 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 software standby mode, but retains its previous value in module 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

13.2.2 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 software standby mode, but retains its previous value in module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	CKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/W	R	R	R	R	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	CMF	0	R/(W)*	Compare Match Flag Indicates whether or not the values of CMCNT and CMCOR match. 0: CMCNT and CMCOR values do not match [Clearing condition] • When 0 is written to CMF after reading CMF = 1 1: CMCNT and CMCOR values match
6	CMIE	0	R/W	Compare Match Interrupt Enable Enables or disables compare match interrupt (CMI) generation when CMCNT and CMCOR values match (CMF = 1). 0: Compare match interrupt (CMI) disabled 1: Compare match interrupt (CMI) enabled
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	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.

13.2.3 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 software standby mode, but retains its previous value in module 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

13.2.4 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 software standby mode, but retains its previous value in module 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

13.3 Operation

13.3.1 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 13.2 shows the operation of the compare match counter.

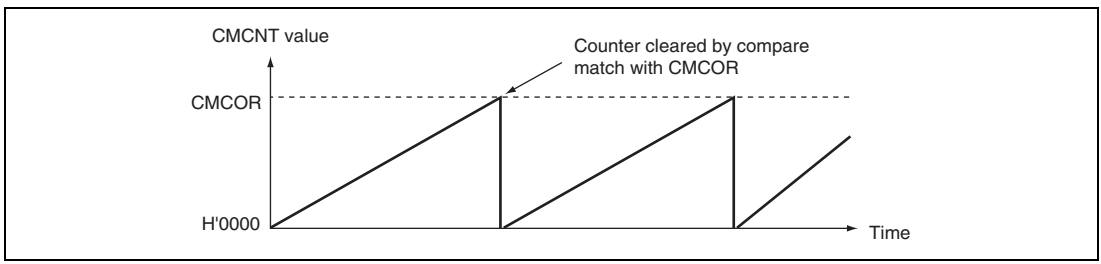


Figure 13.2 Counter Operation

13.3.2 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 13.3 shows the timing.

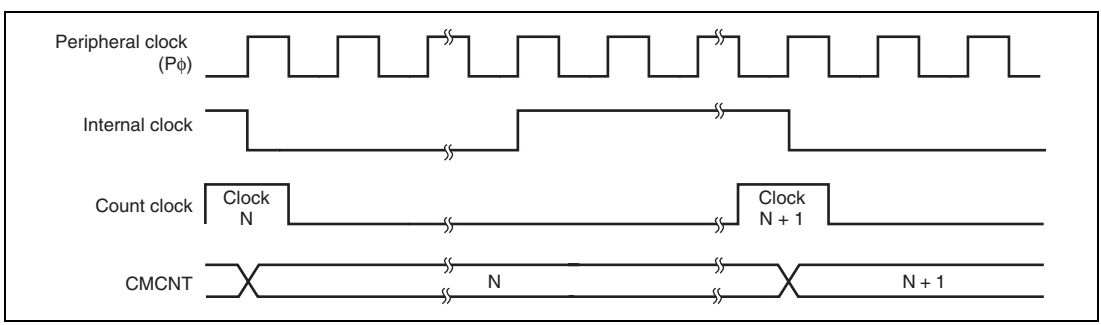


Figure 13.3 Count Timing

13.4 Interrupts

13.4.1 Interrupt Sources and DMA 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.

13.4.2 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 13.4 shows the timing of CMF bit setting.

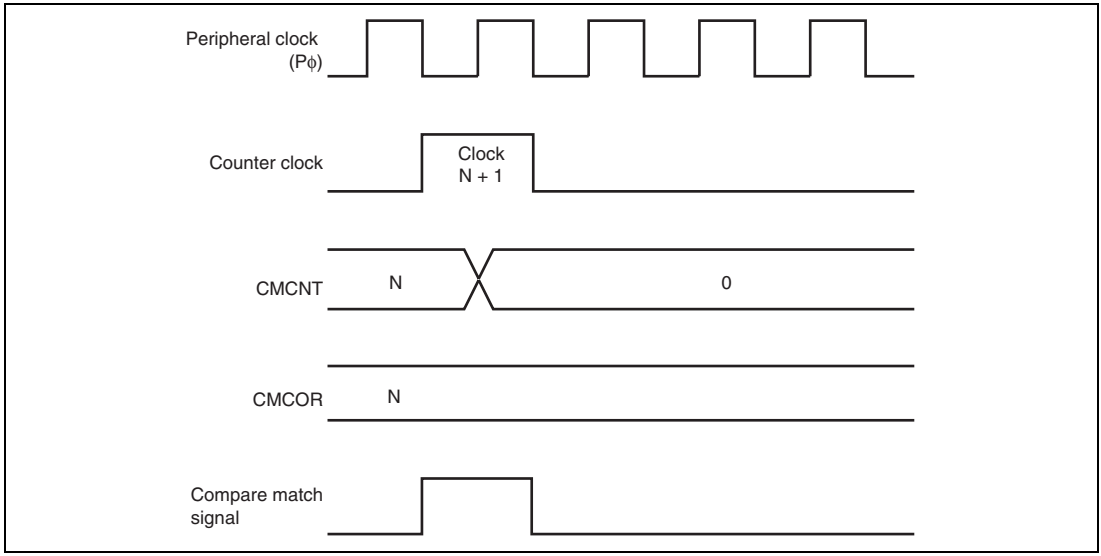


Figure 13.4 Timing of CMF Setting

13.4.3 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.

13.5 Usage Notes

13.5.1 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 13.5 shows the timing to clear the CMCNT counter.

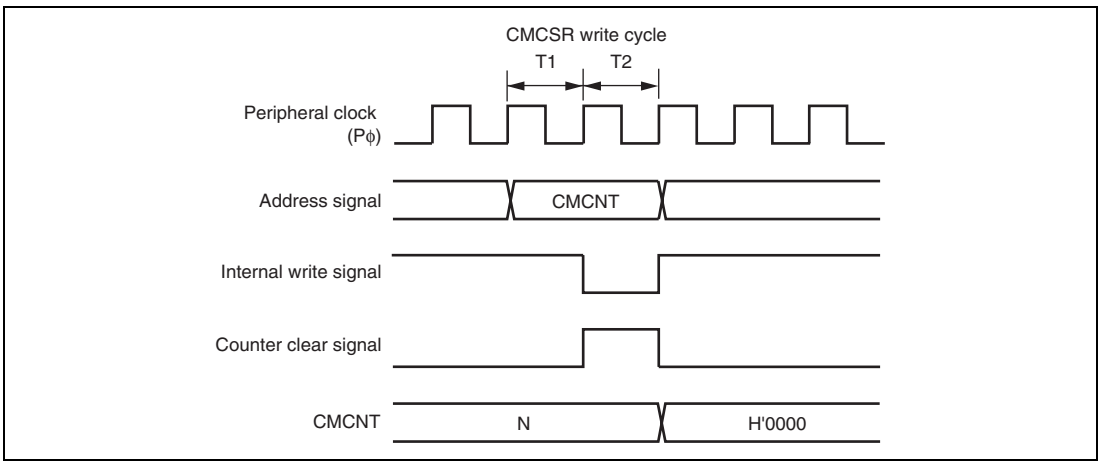


Figure 13.5 Conflict between Write and Compare Match Processes of CMCNT

13.5.2 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 13.6 shows the timing to write to CMCNT in words.

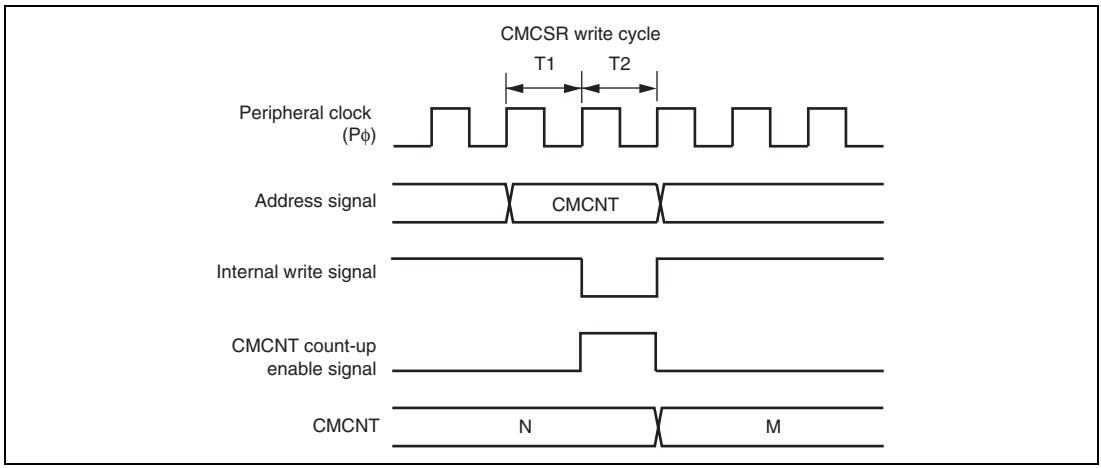


Figure 13.6 Conflict between Word-Write and Count-Up Processes of CMCNT

13.5.3 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 13.7 shows the timing when the count-up occurs in the T2 cycle while writing to CMCNTH in bytes.

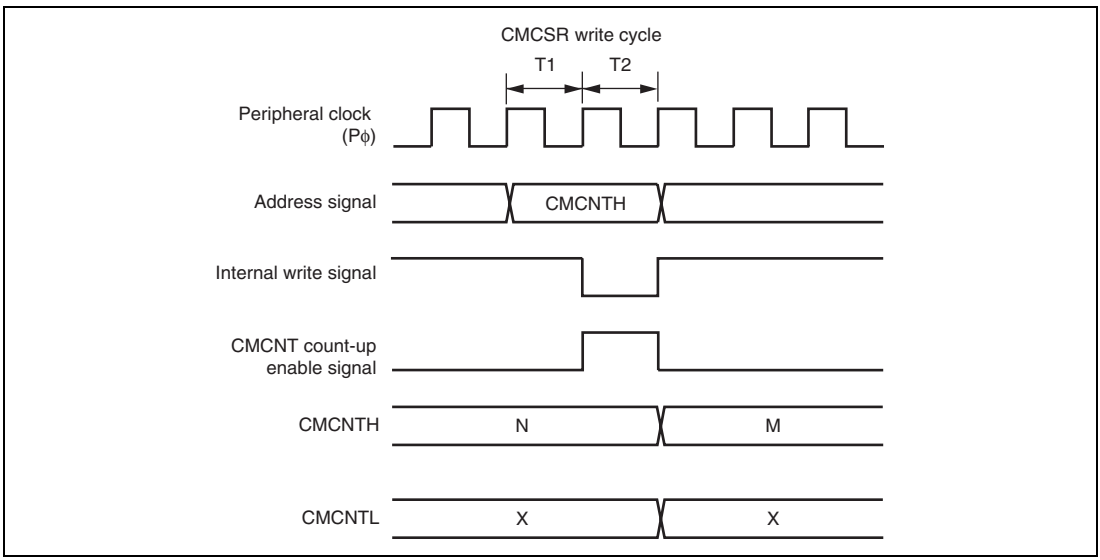


Figure 13.7 Conflict between Byte-Write and Count-Up Processes of CMCNT

13.5.4 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 14 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 software standby mode or 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.

14.1 Features

- Can be used to ensure the clock oscillation settling time
  The WDT is used in leaving software standby mode or 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 (Pϕ × 1 to Pϕ × 1/16384) that are obtained by dividing the peripheral clock can be selected.

Figure 14.1 shows a block diagram of the WDT.

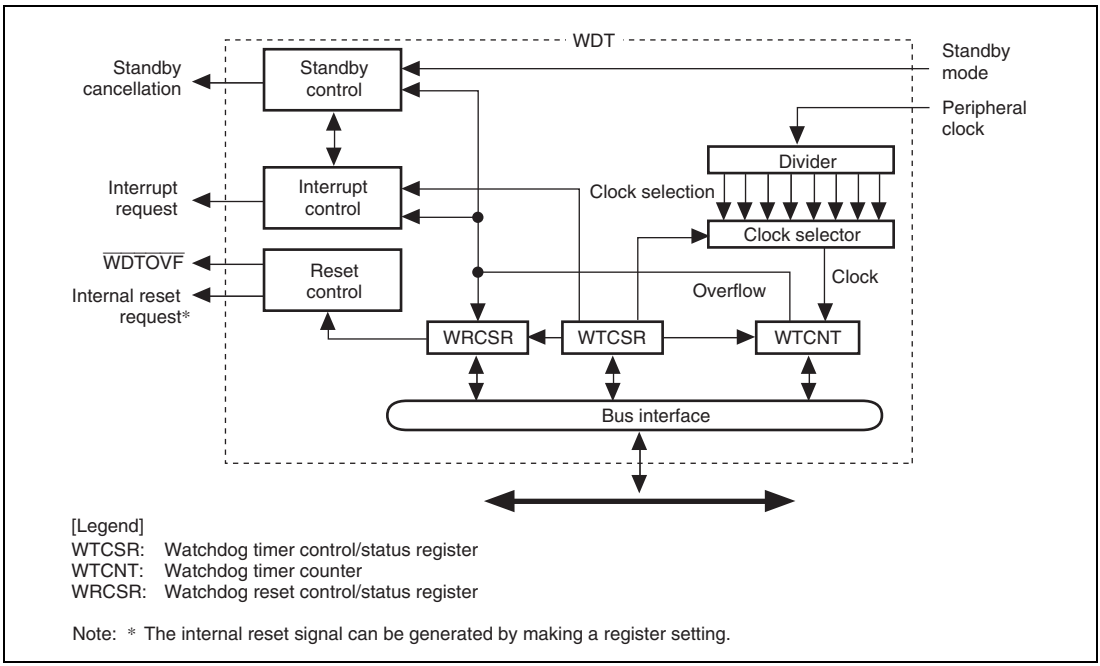


Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pin

Table 14.1 shows the pin configuration of the WDT.

Table 14.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs the counter overflow signal in watchdog timer mode

14.3 Register Descriptions

The WDT has the following registers.

Table 14.2 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*

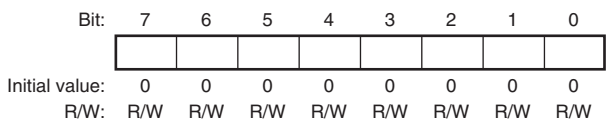
Note: * For the access size, see section 14.3.4, Notes on Register Access.

14.3.1 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 14.3.4, Notes on Register Access, for details.



14.3.2 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. When used to count the clock oscillation settling time for canceling software standby mode, it retains its value after counter overflow.

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 14.3.4, Notes on Register Access, for details.

Bit:	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{\text{IT}}$	TME	-	-	CKS[2:0]		
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)	R/W	R/W	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	IOVF	0	R/(W)	Interval Timer Overflow Indicates that WTCNT has overflowed in interval timer mode. This flag is not set in watchdog timer mode. 0: No overflow 1: WTCNT overflow in interval timer mode [Clearing condition] • When 0 is written to IOVF after reading IOVF
6	WT/$\overline{\text{IT}}$	0	R/W	Timer Mode Select Selects whether to use the WDT as a watchdog timer or an interval timer. 0: Use as interval timer 1: Use as watchdog timer Note: When the WTCNT overflows in watchdog timer mode, the $\overline{\text{WDTOVF}}$ signal is output externally. If this bit is modified when the WDT is running, the up-count may not be performed correctly.

Bit	Bit Name	Initial Value	R/W	Description
5	TME	0	R/W	Timer Enable Starts and stops timer operation. Clear this bit to 0 when using the WDT in software standby mode or when changing the clock frequency. 0: Timer disabled Count-up stops and WTCNT value is retained 1: Timer enabled
4, 3	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
2 to 0	CKS[2:0]	000	R/W	Clock Select These bits select the clock to be used for the WTCNT count from the eight types obtainable by dividing the peripheral clock (Pϕ). The overflow period that is shown in the table is the value when the peripheral clock (Pϕ) is 40 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 × Pϕ	6.4 μs
001:	1/64 × Pϕ	409.6 μs
010:	1/128 × Pϕ	819.2 ms
011:	1/256 × Pϕ	1.64 ms
100:	1/512 × Pϕ	3.3 ms
101:	1/1024 × Pϕ	6.6 ms
110:	1/4096 × Pϕ	26.2 ms
111:	1/16384 × Pϕ	104.9 ms

Note: If bits CKS[2:0] are modified when the WDT is running, the up-count may not be performed correctly. Ensure that these bits are modified only when the WDT is not running.

14.3.3 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 14.3.4, 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.

14.3.4 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 14.2. 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.

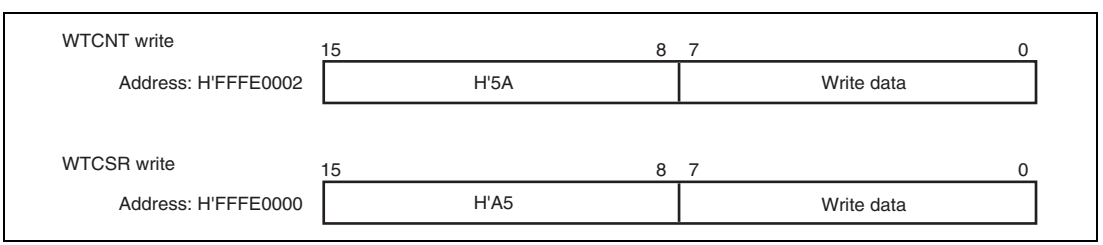


Figure 14.2 Writing to WTCNT and WTCSR

(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 14.3.

To write 0 to the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in 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.

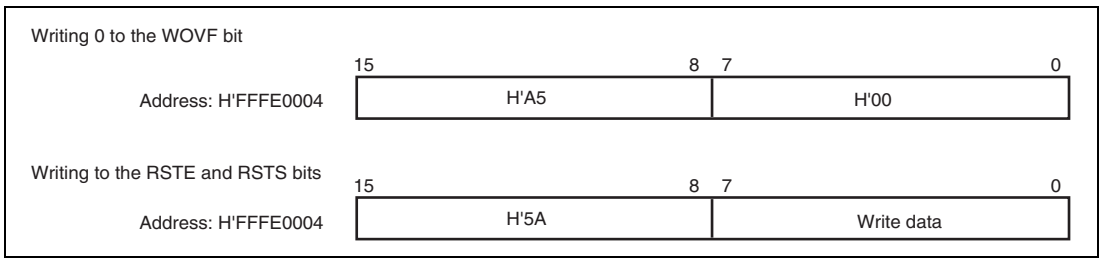


Figure 14.3 Writing to WRCSR

(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.

14.4 WDT Usage

14.4.1 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 23, 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.

14.4.2 Changing the Frequency

To change the frequency used by the PLL, use the WDT. When changing the frequency only by switching the divider, do not use the WDT.

1. Before changing the frequency, 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. When the frequency control register (FRQCR) is written to, this LSI stops temporarily. The WDT starts counting.
4. When the WDT count overflows, the CPG resumes supplying the clock and this LSI resumes operation. The WOVF flag in WRCSR is not set when this happens.
5. The counter stops at the value of H'00.
6. Before changing WTCNT after execution of the frequency change instruction, always confirm that the value of WTCNT is H'00 by reading from WTCNT.

14.4.3 Using Watchdog Timer Mode

1. Set the WT/$\overline{IT}$ 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 $\overline{WDTOVF}$ signal is output externally (figure 14.4). The $\overline{WDTOVF}$ signal can be used to reset the system. The $\overline{WDTOVF}$ signal is output for 64 × Pϕ 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 $\overline{WDTOVF}$ 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 × Pϕ clock cycles.
6. When a WDT overflow reset is generated simultaneously with a reset input on the $\overline{RES}$ pin, the $\overline{RES}$ pin reset takes priority, and the WOVF bit in WRCSR is cleared to 0.

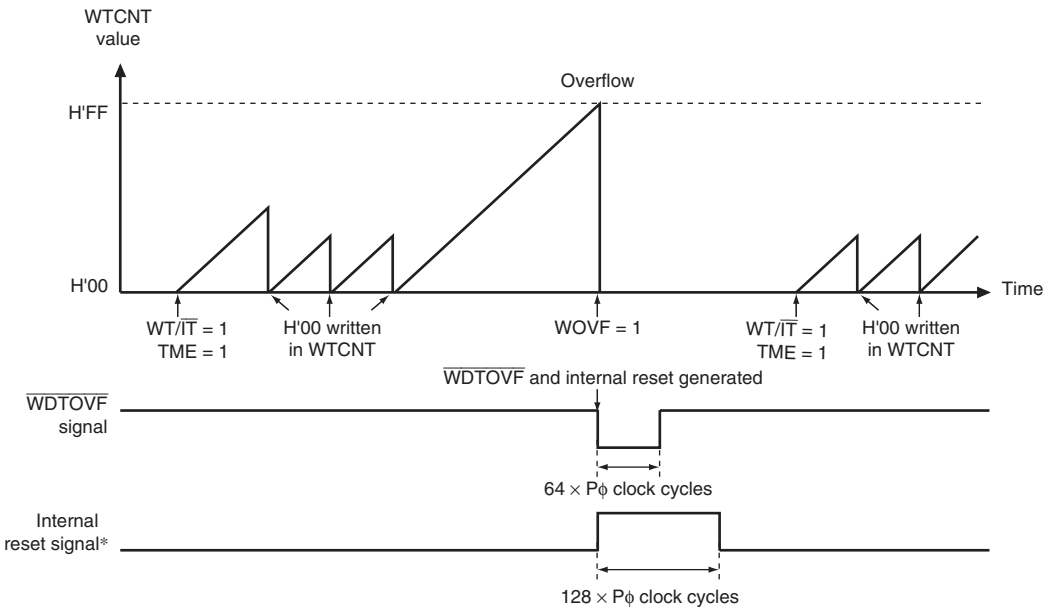


Figure 14.4 Operation in Watchdog Timer Mode

14.4.4 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.

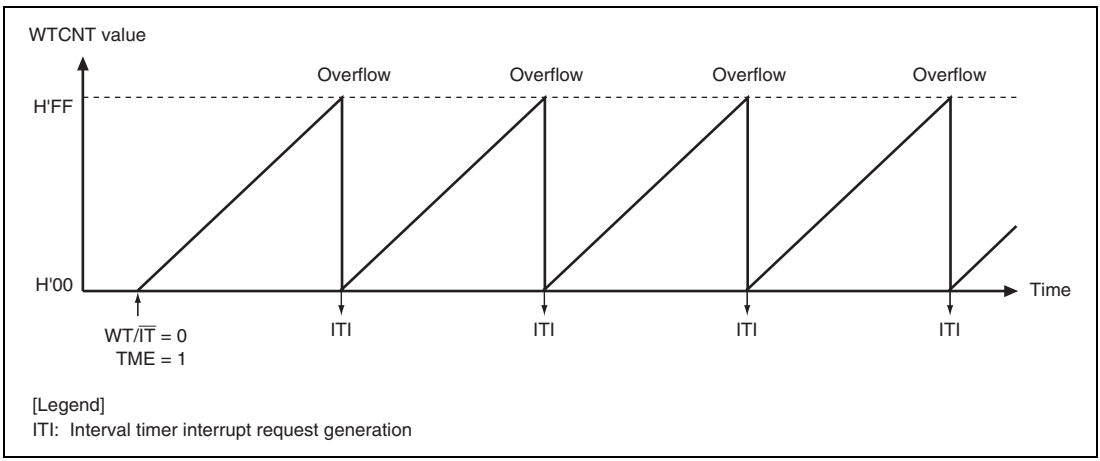


Figure 14.5 Operation in Interval Timer Mode

14.5 Usage Notes

Pay attention to the following points when using the WDT in either the interval timer or watchdog timer mode.

14.5.1 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.

14.5.2 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.

14.5.3 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 14.6.

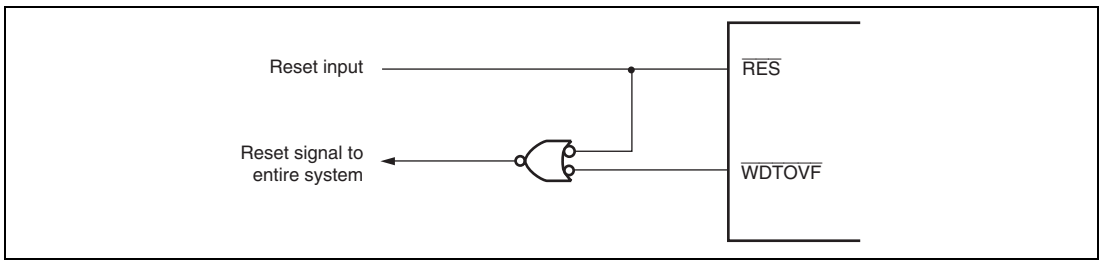


Figure 14.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

14.5.4 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.

However, if the duration from generation of the manual reset to the bus cycle end is equal to or longer than the duration of the internal manual reset activated, the occurrence of the internal manual reset source is ignored instead of being pended, and the manual reset exception handling is not executed.

Section 15 Serial Communication Interface with FIFO (SCIF)

This LSI has a four-channel 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.

15.1 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 15.1 shows a block diagram of the SCIF.

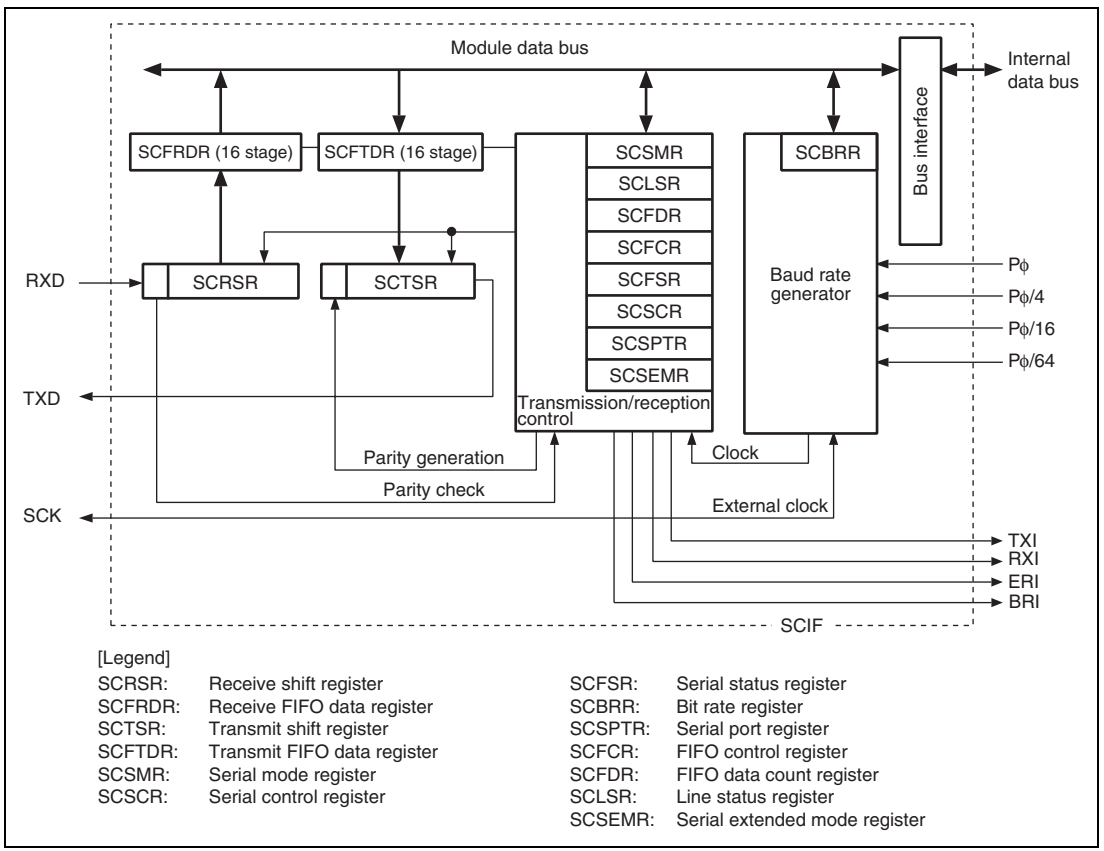


Figure 15.1 Block Diagram of SCIF

15.2 Input/Output Pins

Table 15.1 shows the pin configuration of the SCIF.

Table 15.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 3	Serial clock pins	SCK0 to SCK3	I/O	Clock I/O
	Receive data pins	RXD0 to RXD3	Input	Receive data input
	Transmit data pins	TXD0 to TXD3	Output	Transmit data output

15.3 Register Descriptions

The SCIF has the following registers.

Table 15.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'0000	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	R/W	H'FF	H'FFFE8004	8
	Serial control register_0	SCSCR_0	R/W	H'0000	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	W	Undefined	H'FFFE800C	8
	Serial status register_0	SCFSR_0	R/(W)*[1]	H'0060	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	R	Undefined	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	R/W	H'0000	H'FFFE8018	16
	FIFO data count register_0	SCFDR_0	R	H'0000	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	R/W	H'0050	H'FFFE8020	16
	Line status register_0	SCLSR_0	R/(W)*[2]	H'0000	H'FFFE8024	16
1	Serial mode register_1	SCSMR_1	R/W	H'0000	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	R/W	H'FF	H'FFFE8804	8
	Serial control register_1	SCSCR_1	R/W	H'0000	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	W	Undefined	H'FFFE880C	8
	Serial status register_1	SCFSR_1	R/(W)*[1]	H'0060	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	R	Undefined	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	R/W	H'0000	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	R	H'0000	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	R/W	H'0050	H'FFFE8820	16
	Line status register_1	SCLSR_1	R/(W)*[2]	H'0000	H'FFFE8824	16
	Serial extended mode register_1	SCSEMR_1	R/W	H'00	H'FFFE8900	8

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	Serial mode register_2	SCSMR_2	R/W	H'0000	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	R/W	H'FF	H'FFFE9004	8
	Serial control register_2	SCSCR_2	R/W	H'0000	H'FFFE9008	16
	Transmit FIFO data register_2	SCFTDR_2	W	Undefined	H'FFFE900C	8
	Serial status register_2	SCFSR_2	R/(W)*1	H'0060	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	R	Undefined	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	R/W	H'0000	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	R	H'0000	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	R/W	H'0050	H'FFFE9020	16
	Line status register_2	SCLSR_2	R/(W)*2	H'0000	H'FFFE9024	16
	Serial extended mode register_2	SCSEMR_2	R/W	H'00	H'FFFE9100	8
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'0050	H'FFFE9820	16
	Line status register_3	SCLSR_3	R/(W)*2	H'0000	H'FFFE9824	16

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.

15.3.1 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:	-	-	-	-	-	-	-	-

15.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to an undefined value by a power-on reset.

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

15.3.3 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:	-	-	-	-	-	-	-	-

15.3.4 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

15.3.5 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 15.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

15.3.6 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 to1. 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 to1, receive-error (ERI) interrupts requested when the ER flag in SCFSR is set to1, and break (BRI) interrupts requested when the BRK flag in SCFSR or the ORER flag in line status register (SCLSR) is set to1. 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

15.3.7 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.

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. 1: The quantity of transmit data in SCFTDR is equal to or 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 equal to or 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. 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. 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.

15.3.8 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. Each channel has independent baud rate generator control, so different values can be set in four channels.

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:
  (1) In normal mode (when the ABCS bit in SCSEMR is 0)

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

  (2) In serial extended mode (when the ABCS bit in 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 15.3.)

Table 15.3 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:

(1) In normal mode (when the ABCS bit in SCSEMR is 0)

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

(2) In serial extended mode (when the ABCS bit in 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 15.4 lists examples of SCBRR settings in asynchronous mode, and table 15.5 lists examples of SCBRR settings in clocked synchronous mode.

Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode)

	Pϕ (MHz)								
	32			36			40		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	141	0.03	3	159	–0.12	3	117	–0.25
150	3	103	0.16	3	116	0.16	3	129	0.16
300	3	51	0.16	3	58	–0.69	3	64	0.16
600	2	103	0.16	2	116	0.16	2	129	0.16
1200	2	51	0.16	2	58	–0.69	2	64	0.16
2400	1	103	0.16	1	116	0.16	1	129	0.16
4800	1	51	0.16	1	58	–0.69	1	64	0.16
9600	0	103	0.16	0	116	0.16	0	129	0.16
19200	0	51	0.16	0	58	–0.69	0	64	0.16
31250	0	31	0.00	0	35	0.00	0	39	0.00
38400	0	25	0.16	0	28	1.02	0	32	–1.36

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

Table 15.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode)

	Pϕ (MHz)					
	32		36		40	
Bit Rate (bit/s)	**n**	**N**	**n**	**N**	**n**	**N**
500	—	—				
1 k	3	124	3	140	3	155
2.5 k	2	199	2	224	2	249
5 k	2	99	2	112	2	124
10 k	2	49	2	55	2	62
25 k	1	79	1	89	1	97
50 k	1	39	1	44	1	48
100 k	0	79	0	89	0	97
250 k	0	31	0	35	0	38
500 k	0	15	0	17	0	19
1 M	0	7	0	8	0	9
2 M	0	3	—	—	0	4

[Legend]

Blank: No setting possible

—: Setting possible, but error occurs

Table 15.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 15.7 and 15.8 list the maximum bit rates when the external clock input is used.

Note: * Make sure that the electrical characteristics of this LSI and that of a connected LSI are satisfied.

Table 15.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
32	1000000	0	0
36	1125000	0	0
40	1250000	0	0

Table 15.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
32	8.0000	500000
36	9.0000	562500
40	10.0000	625000

Table 15.8 Maximum Bit Rates with External Clock Input (Clocked Synchronous Mode, $t_{Scyc} = 12t_{pcyc}$)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
32	2.6667	2666666.7
36	3.0000	3000000.0
40	3.3333	3333333.3

Note: Confirm that these bit rates meet the electrical characteristics of this LSI and the remote communication device.

15.3.9 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

15.3.10 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.

15.3.11 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	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	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
2	SCKDT	0	R/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

Bit	Bit Name	Initial Value	R/W	Description
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	0	R/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

15.3.12 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.

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

15.3.13 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. The basic clock can be set for only channels 1 and 2.

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.

15.4 Operation

15.4.1 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 15.9. 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 15.10.

(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 15.9 SCSMR Settings and SCIF Communication Formats

SCSMR Settings					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 15.10 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR	SCSCR		SCIF Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0		Clock Source	SCK Pin Function
C/$\overline{A}$	CKE1	CKE0	Mode		
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

15.4.2 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 15.2 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.

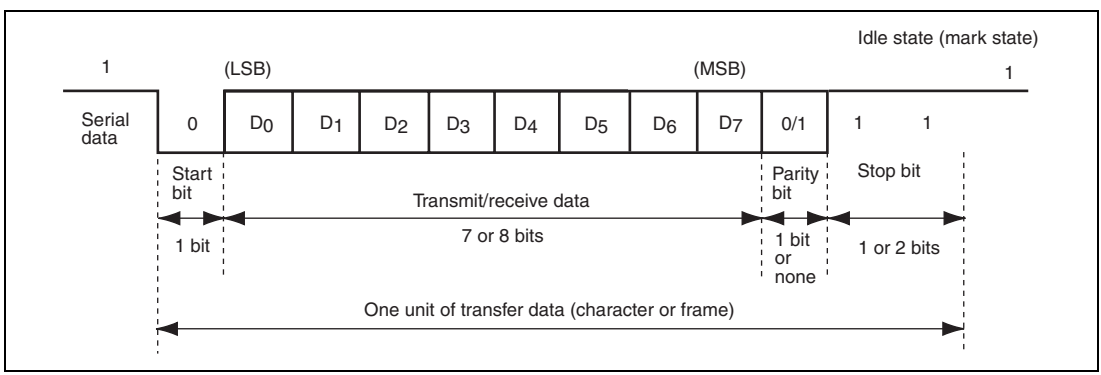


Figure 15.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

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.

(1) Transmit/Receive Formats

Table 15.11 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 15.11 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 15.10, SCSMR and SCSCR Settings and SCIF Clock Source Selection.

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 15.3 shows a sample flowchart for initializing the SCIF.

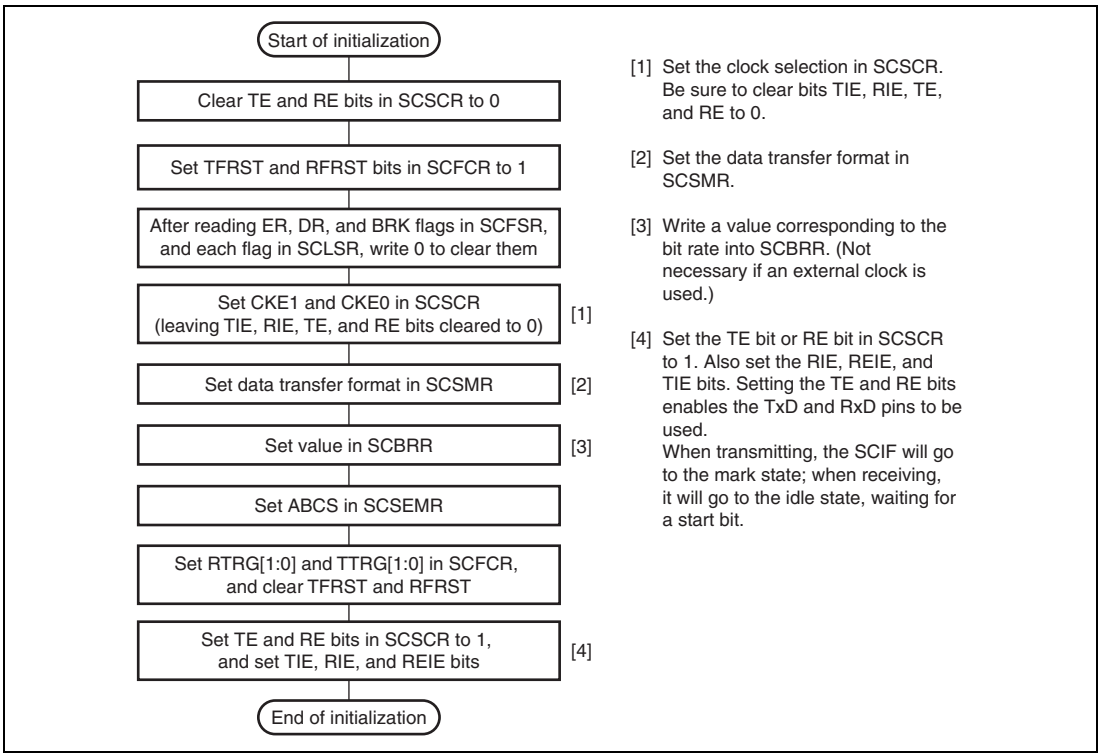


Figure 15.3 Sample Flowchart for SCIF Initialization

- **Transmitting Serial Data (Asynchronous Mode)**

Figure 15.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

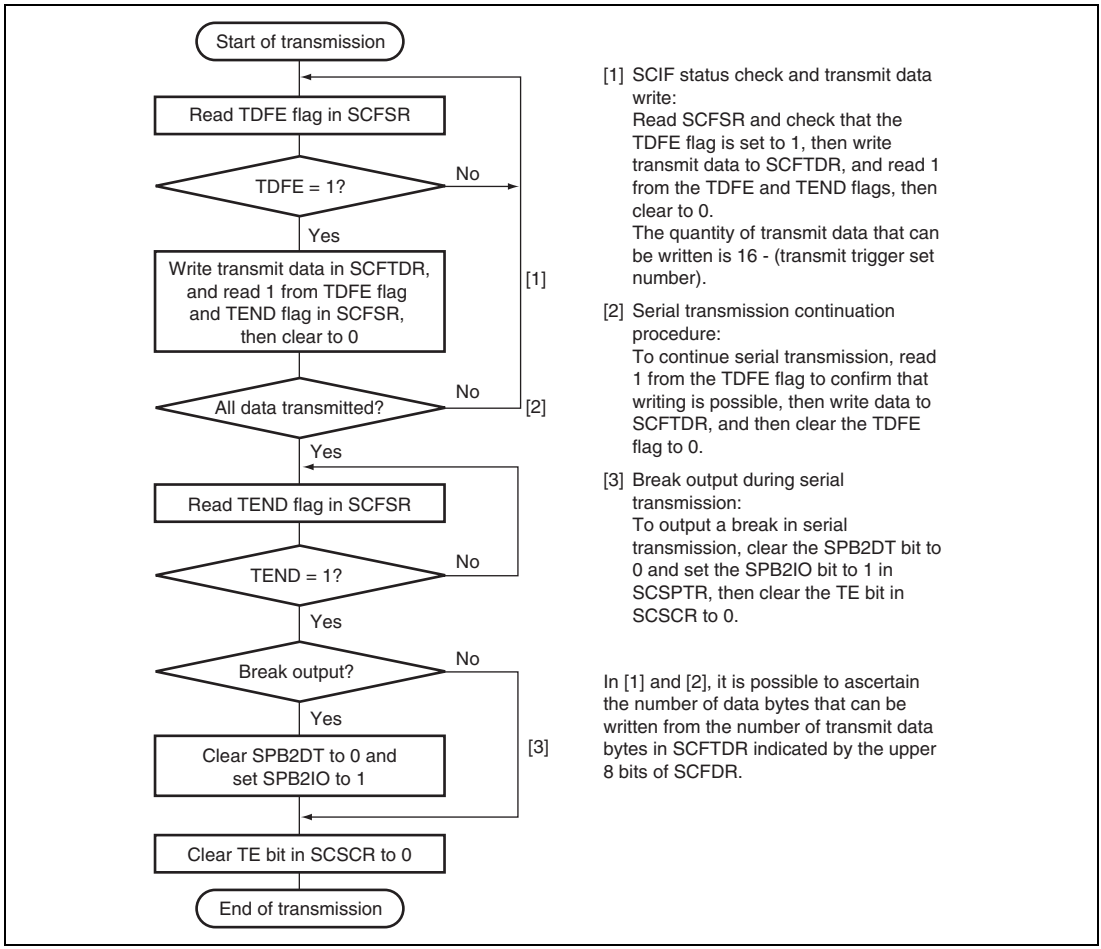


Figure 15.4 Sample Flowchart for Transmitting Serial Data

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 15.5 shows an example of the operation for transmission.

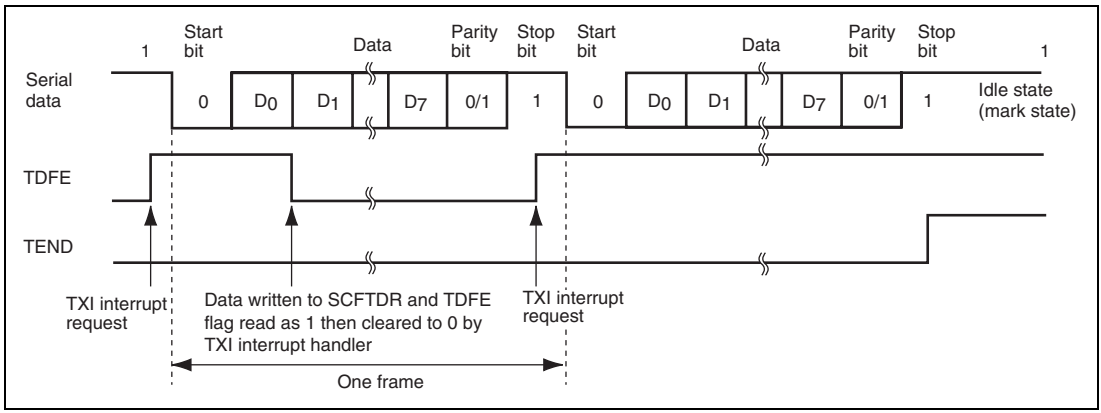


Figure 15.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

- **Receiving Serial Data (Asynchronous Mode)**

Figures 15.6 and 15.7 show sample flowcharts for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

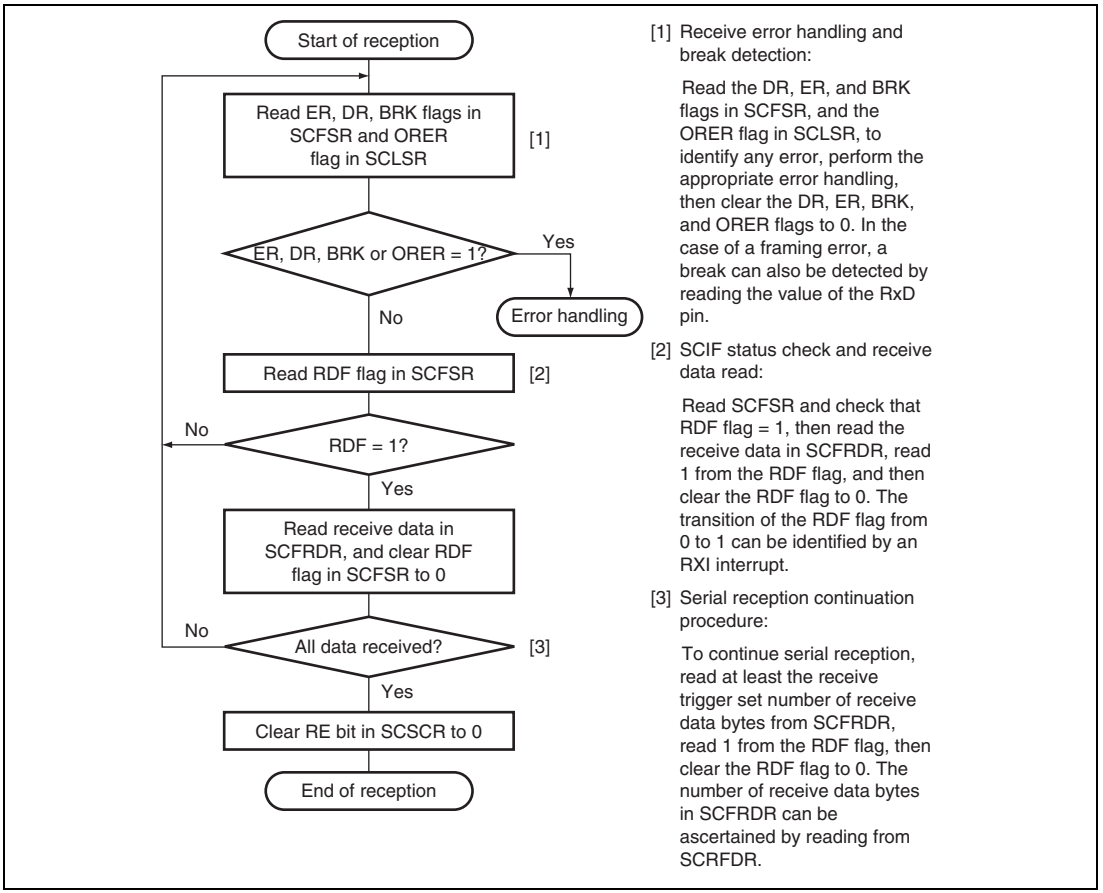
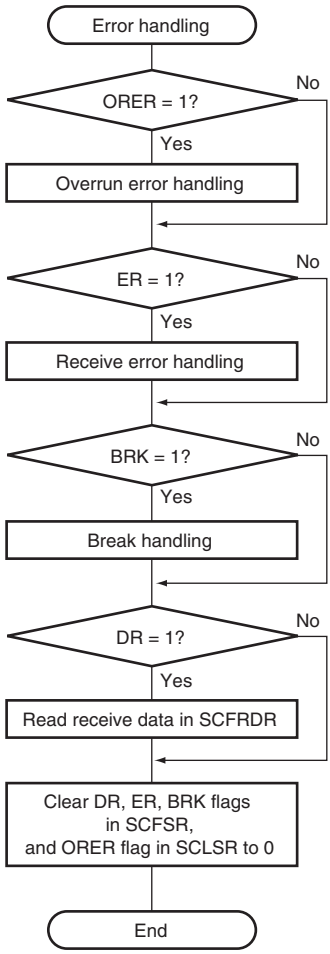


Figure 15.6 Sample Flowchart for Receiving Serial Data



- 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 15.7 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 15.8 shows an example of the operation for reception.

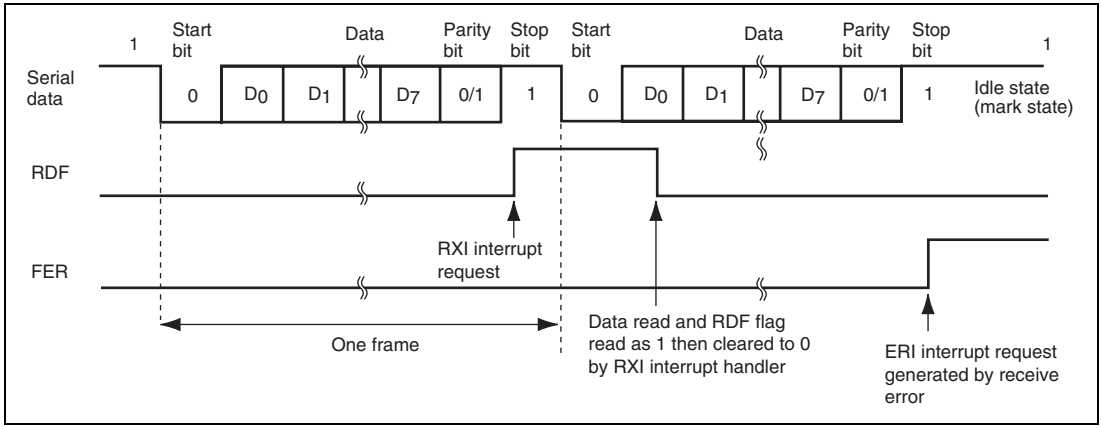


Figure 15.8 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

15.4.3 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 15.9 shows the general format in clocked synchronous serial communication.

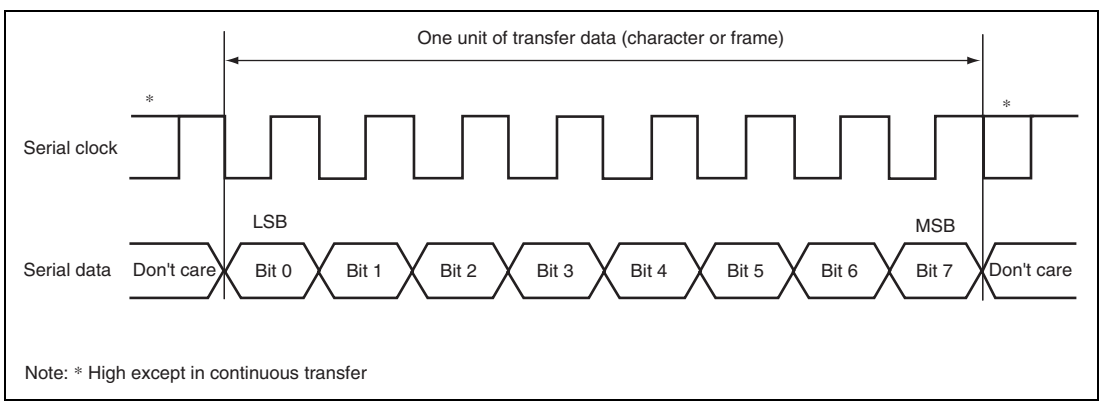


Figure 15.9 Data Format in Clocked Synchronous Communication

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 15.10 shows a sample flowchart for initializing the SCIF.

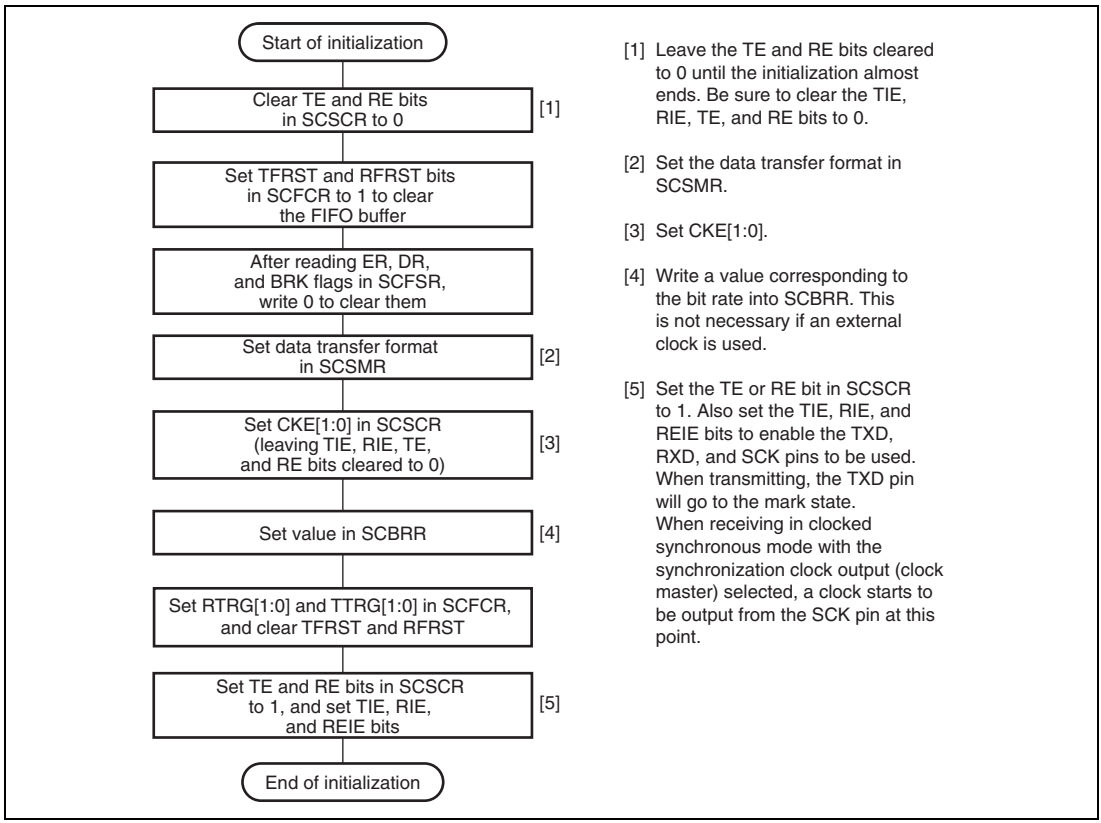


Figure 15.10 Sample Flowchart for SCIF Initialization

- **Transmitting Serial Data (Clocked Synchronous Mode)**

Figure 15.11 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

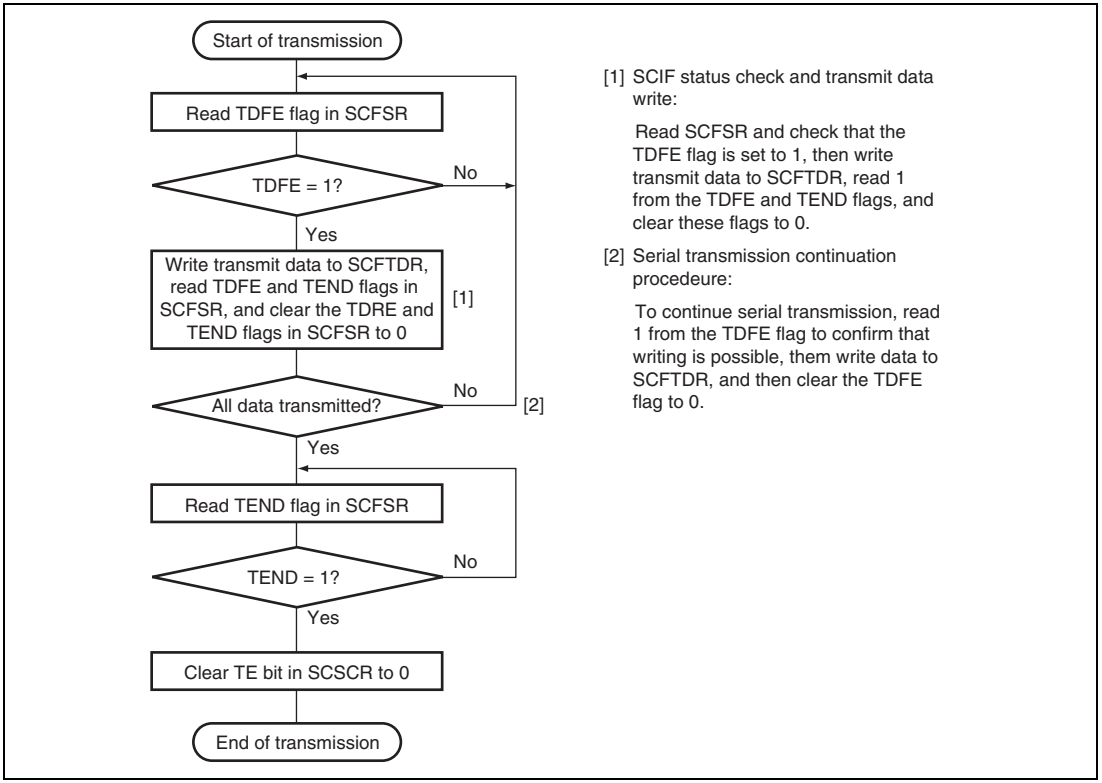


Figure 15.11 Sample Flowchart for Transmitting Serial Data

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 15.12 shows an example of SCIF transmit operation.

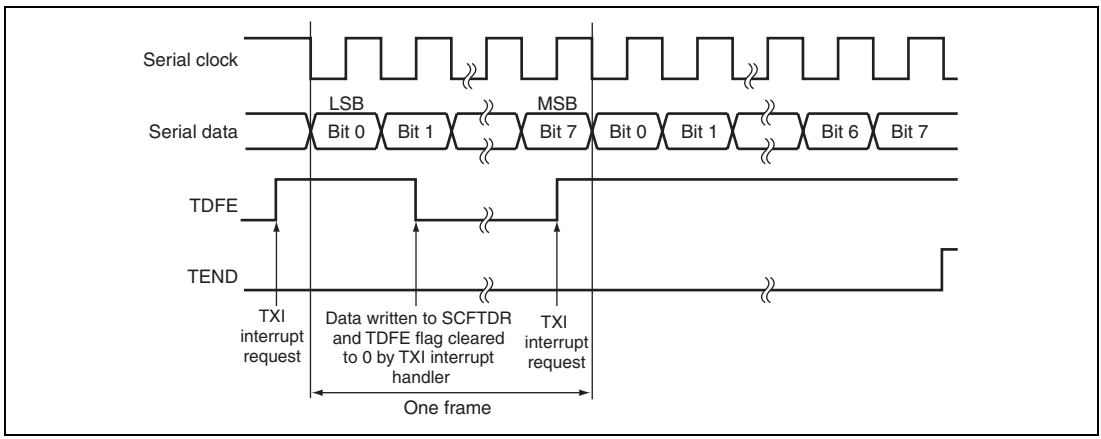


Figure 15.12 Example of SCIF Transmit Operation

- **Receiving Serial Data (Clocked Synchronous Mode)**

Figures 15.13 and 15.14 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.

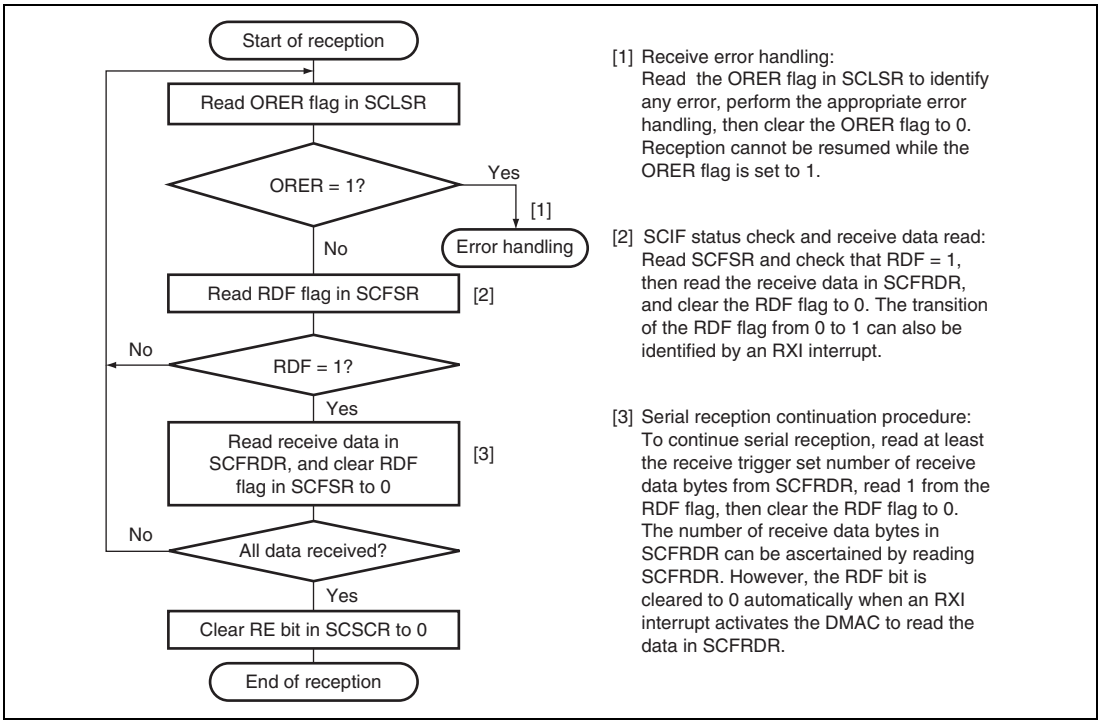


Figure 15.13 Sample Flowchart for Receiving Serial Data (1)

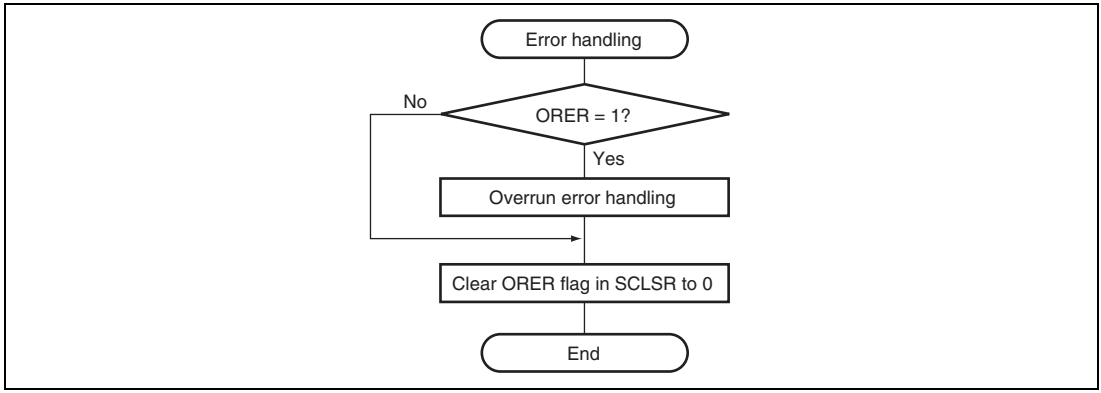


Figure 15.14 Sample Flowchart for Receiving Serial Data (2)

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 15.15 shows an example of SCIF receive operation.

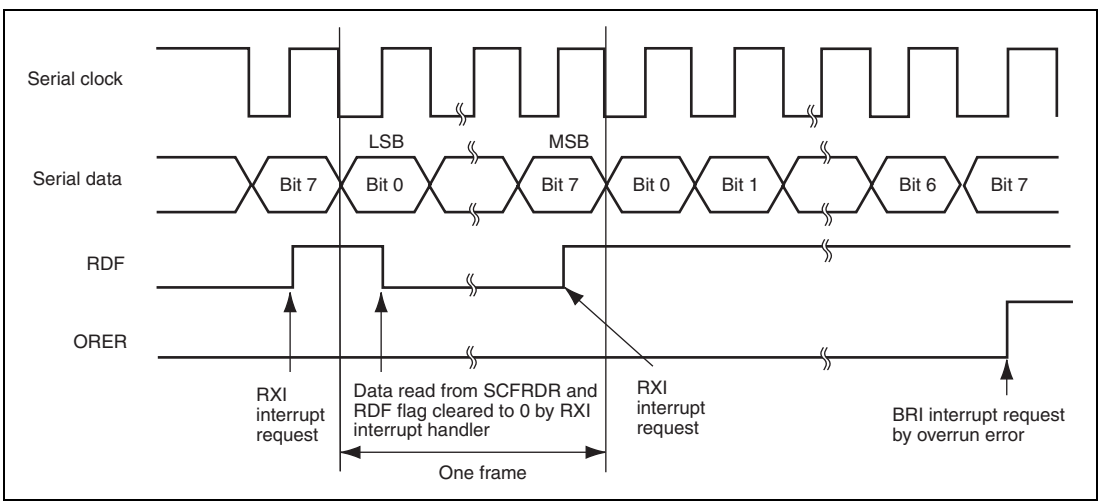


Figure 15.15 Example of SCIF Receive Operation

- **Transmitting and Receiving Serial Data Simultaneously (Clocked Synchronous Mode)**

Figure 15.16 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.

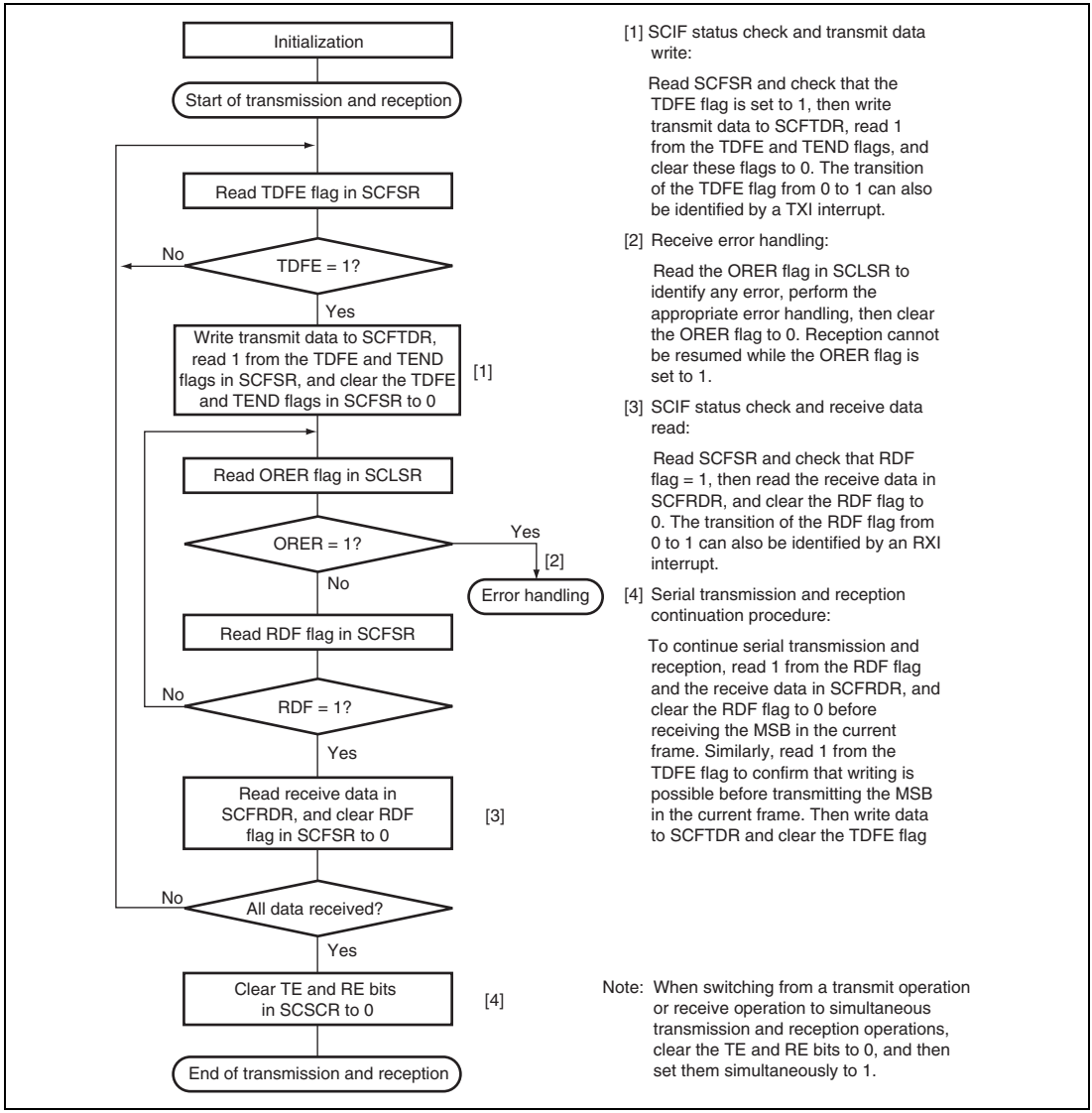


Figure 15.16 Sample Flowchart for Transmitting/Receiving Serial Data

15.5 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 15.12 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 can be activated and data transfer performed by this TXI interrupt request. At this time, 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 can be activated and data transfer performed by this RXI interrupt request. At this time, 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 15.12 SCIF Interrupt Sources

Interrupt Source	Description	DMAC 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

15.6 Usage Notes

Note the following when using the SCIF.

15.6.1 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).

15.6.2 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).

15.6.3 Restriction on DMAC Usage

When the DMAC 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.

15.6.4 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.

15.6.5 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.

15.6.6 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 15.17.

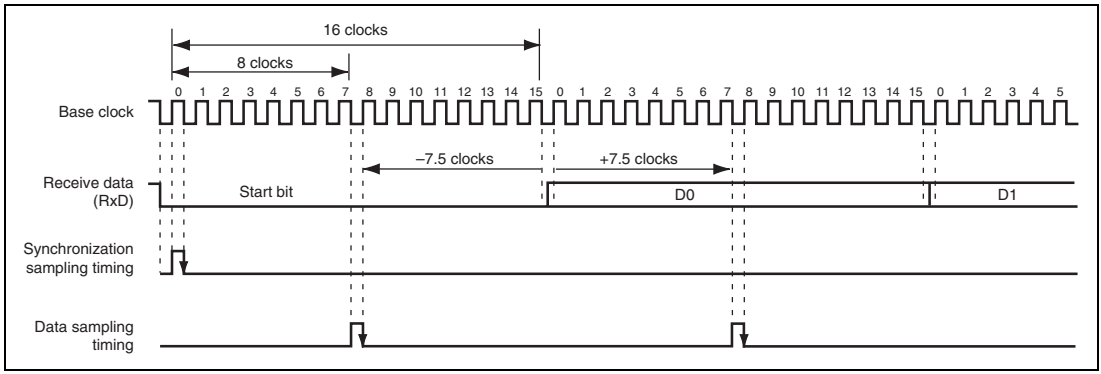


Figure 15.17 Receive Data Sampling Timing in Asynchronous Mode

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.

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 when ABCS = 0, and N = 8 when ABCS = 1)
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, D = 0.5, and N = 16, 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%.

15.6.7 FER and PER Flags in the Serial Status Register (SCFSR)

The FER (framing error) and PER (parity error) flags in the serial status register (SCFSR) are status flags of the receive FIFO data register (SCFRDR) to be read next. If the CPU or DMAC reads the receive FIFO data register, the FER (framing error) and PER (parity error) flags of the current receive data will be lost. To check the framing error and parity error status of the current receive data correctly, the serial status register (SCFSR) should be read before the receive FIFO data register is read.

Section 16 I²C Bus Interface 3 (IIC3)

The I²C bus interface 3 conforms to and provides a subset of the Philips I²C (Inter-IC) bus interface functions. However, the configuration of the registers that control the I²C bus differs partly from the Philips register configuration.

16.1 Features

- Selection of I²C format or clocked synchronous serial format
- Continuous transmission/reception

  Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.

I²C bus format:

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

  In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically. If transmission/reception is not yet possible, set the SCL to low until preparations are completed.
- Six interrupt sources

  Transmit data empty (including slave-address match), transmit end, receive data full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- The direct memory access controller (DMAC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.
- Direct bus drive

  Two pins, SCL and SDA pins, function as NMOS open-drain outputs when the bus drive function is selected.

Clocked synchronous serial format:

- Four interrupt sources

  Transmit-data-empty, transmit-end, receive-data-full, and overrun error
- The direct memory access controller (DMAC) can be activated by a transmit-data-empty request or receive-data-full request to transfer data.

Figure 16.1 shows a block diagram of the I²C bus interface 3.

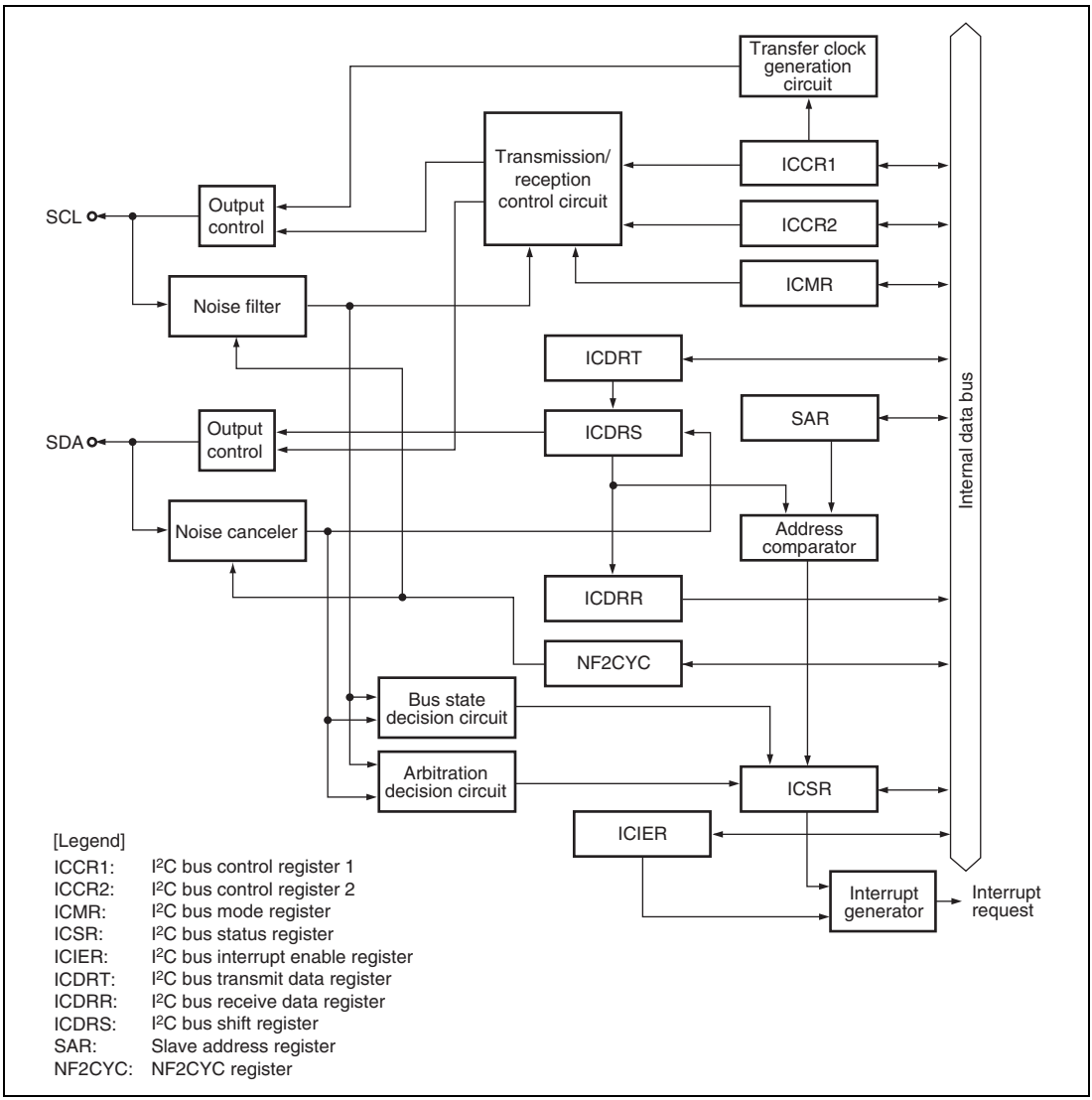


Figure 16.1 Block Diagram of I²C Bus Interface 3

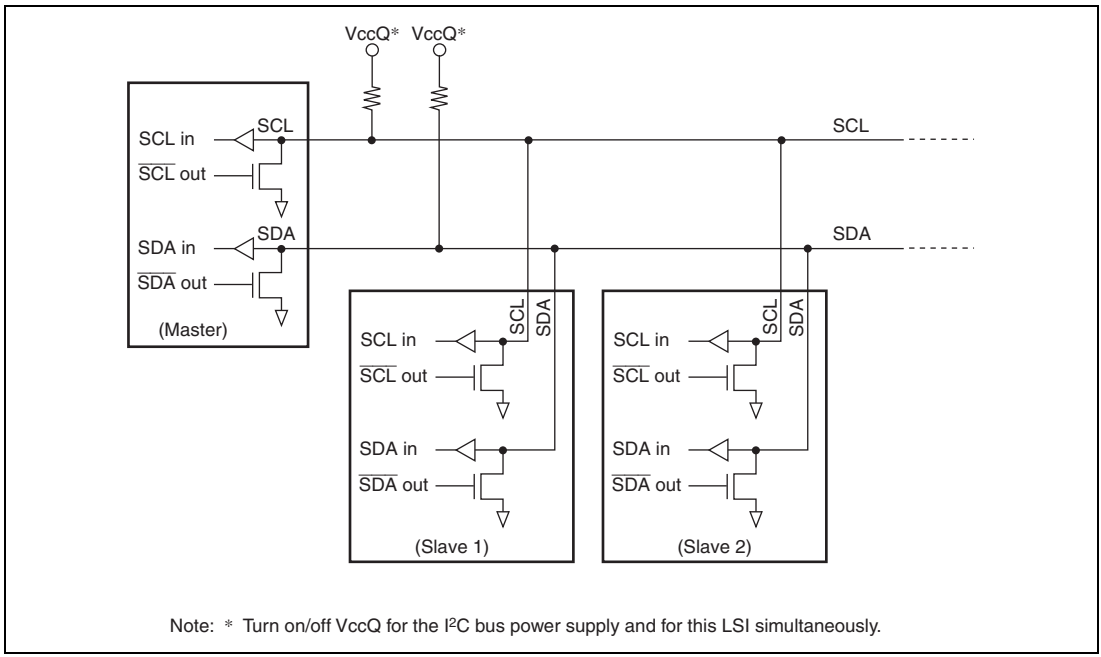
16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the I²C bus interface 3.

Table 16.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Serial clock	SCL	I/O	I²C serial clock input/output
Serial data	SDA	I/O	I²C serial data input/output

Figure 16.2 shows an example of I/O pin connections to external circuits.



Note: * Turn on/off VccQ for the I²C bus power supply and for this LSI simultaneously.

Figure 16.2 External Circuit Connections of I/O Pins

16.3 Register Descriptions

The I²C bus interface 3 has the following registers.

Table 16.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
I²C bus control register 1	ICCR1	R/W	H'00	H'FFFEE000	8
I²C bus control register 2	ICCR2	R/W	H'7D	H'FFFEE001	8
I²C bus mode register	ICMR	R/W	H'38	H'FFFEE002	8
I²C bus interrupt enable register	ICIER	R/W	H'00	H'FFFEE003	8
I²C bus status register	ICSR	R/W	H'00	H'FFFEE004	8
Slave address register	SAR	R/W	H'00	H'FFFEE005	8
I²C bus transmit data register	ICDRT	R/W	H'FF	H'FFFEE006	8
I²C bus receive data register	ICDRR	R/W	H'FF	H'FFFEE007	8
NF2CYC register	NF2CYC	R/W	H'00	H'FFFEE008	8

16.3.1 I²C Bus Control Register 1 (ICCR1)

ICCR1 is an 8-bit readable/writable register that enables or disables the I²C bus interface 3, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

ICCR1 is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I²C Bus Interface 3 Enable 0: This module is halted. (SCL and SDA pins function as ports.) 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable Enables or disables continuous reception when TRS = 0 and ICDRR is not read. If ICDRR cannot be read by the rising of 8th clock cycle of SCL in master receive mode, reception in byte units should be performed by setting the RCVD bit to 1. 0: Enables continuous reception 1: Disables continuous reception

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. When seven bits after the start condition is issued in slave receive mode match the slave address set to SAR and the 8th bit is set to 1, TRS is automatically set to 1. If an overrun error occurs in master receive mode with the clocked synchronous serial format, MST is cleared and the mode changes to slave receive mode. Operating modes are described below according to MST and TRS combination. When clocked synchronous serial format is selected and MST = 1, clock is output. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3 to 0	CKS[3:0]	0000	R/W	Transfer Clock Select These bits should be set according to the necessary transfer rate (table 16.3) in master mode.

Table 16.3 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate (kHz)		
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 32MHz (128/4)	Pϕ = 36MHz (144/4)	Pϕ = 40MHz (160/4)
0	0	0	0	Pϕ/44	727.3	818.2	909.1
			1	Pϕ/52	615.4	692.3	769.2
		1	0	Pϕ/64	500.0	562.5	625.0
			1	Pϕ/72	444.4	500.0	555.6
	1	0	0	Pϕ/84	381.0	428.6	476.2
			1	Pϕ/92	347.8	391.3	434.8
		1	0	Pϕ/100	320.0	360.0	400.0
			1	Pϕ/108	296.3	333.3	370.4
1	0	0	0	Pϕ/176	181.8	204.5	227.3
			1	Pϕ/208	153.8	173.1	192.3
		1	0	Pϕ/256	125.0	140.6	156.3
			1	Pϕ/288	111.1	125.0	138.9
	1	0	0	Pϕ/336	95.2	107.1	119.0
			1	Pϕ/368	87.0	97.8	108.7
		1	0	Pϕ/400	80.0	90.0	100.0
			1	Pϕ/432	74.1	83.3	92.6

Note: The settings should satisfy external specifications.

16.3.2 I²C Bus Control Register 2 (ICCR2)

ICCR2 is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in the control part of the I²C bus.

ICCR2 is initialized to H'7D by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	-	IICRST	-
Initial value:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start/stop conditions in master mode. With the clocked synchronous serial format, this bit is always read as 0. With the I²C bus format, this bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 in BBSY and 0 in SCP to issue a stop condition.
6	SCP	1	R/W	Start/Stop Issue Condition Disable Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. Even if 1 is written to this bit, the data will not be stored.

Bit	Bit Name	Initial Value	R/W	Description
5	SDAO	1	R/W	SDA Output Value Control This bit is used with SDAOP when modifying output level of SDA. This bit should not be manipulated during transfer. 0: When reading, SDA pin outputs low. When writing, SDA pin is changed to output low. 1: When reading, SDA pin outputs high. When writing, SDA pin is changed to output Hi-Z (outputs high by external pull-up resistance).
4	SDAOP	1	R/W	SDAO Write Protect Controls change of output level of the SDA pin by modifying the SDAO bit. To change the output level, clear SDAO and SDAOP to 0 or set SDAO to 1 and clear SDAOP to 0. This bit is always read as 1.
3	SCLO	1	R	SCL Output Level Monitors SCL output level. When SCLO is 1, SCL pin outputs high. When SCLO is 0, SCL pin outputs low.
2	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
1	IICRST	0	R/W	IIC Control Part Reset Resets the control part except for I²C registers. If this bit is set to 1 when hang-up occurs because of communication failure during I²C bus operation, some IIC3 registers and the control part can be reset.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

16.3.3 I²C Bus Mode Register (ICMR)

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the transfer bit count.

ICMR is initialized to H'38 by a power-on reset. Bits BC[2:0] are initialized to H'0 by the IICRST bit in ICCR2.

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

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	BCWP	1	R/W	BC Write Protect Controls the BC[2:0] modifications. When modifying the BC[2:0] bits, this bit should be cleared to 0. In clocked synchronous serial mode, the BC[2:0] bits should not be modified. 0: When writing, values of the BC[2:0] bits are set. 1: When reading, 1 is always read. When writing, settings of the BC[2:0] bits are invalid.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I²C bus format, the data is transferred with one addition acknowledge bit. Should be made between transfer frames. If these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. After the stop condition is detected, the value of these bits returns automatically to B'111. The value returns to B'000 at the end of a data transfer, including the acknowledge bit. These bits are cleared by a power-on reset and in software standby mode and module standby mode. These bits are also cleared by setting the IICRST bit of ICCR2 to 1. With the clocked synchronous serial format, these bits should not be modified. I²C Bus Format / Clocked Synchronous Serial Format 000: 9 bits / 000: 8 bits 001: 2 bits / 001: 1 bit 010: 3 bits / 010: 2 bits 011: 4 bits / 011: 3 bits 100: 5 bits / 100: 4 bits 101: 6 bits / 101: 5 bits 110: 7 bits / 110: 6 bits 111: 8 bits / 111: 7 bits

16.3.4 I²C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits received.

ICIER is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) and the overrun error interrupt request (ERI) in the clocked synchronous format when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection interrupt request (NAKI) and the overrun error (OVE set in ICSR) interrupt request (ERI) in the clocked synchronous format when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.
3	STIE	0	R/W	Stop Condition Detection Interrupt Enable Enables or disables the stop condition detection interrupt request (STPI) when the STOP bit in ICSR is set. 0: Stop condition detection interrupt request (STPI) is disabled. 1: Stop condition detection interrupt request (STPI) is enabled.
2	ACKE	0	R/W	Acknowledge Bit Judgment Select 0: The value of the receive acknowledge bit is ignored, and continuous transfer is performed. 1: If the receive acknowledge bit is 1, continuous transfer is halted.
1	ACKBR	0	R	Receive Acknowledge In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified. This bit can be canceled by setting the BBSY bit in ICCR2 to 1. 0: Receive acknowledge = 0 1: Receive acknowledge = 1
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, this bit specifies the bit to be sent at the acknowledge timing. 0: 0 is sent at the acknowledge timing. 1: 1 is sent at the acknowledge timing.

16.3.5 I²C Bus Status Register (ICSR)

ICSR is an 8-bit readable/writable register that confirms interrupt request flags and their status.

ICSR is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When data is written to ICDRT [Setting conditions] • When data is transferred from ICDRT to ICDRS and ICDRT becomes empty • When TRS is set • When the start condition (including retransmission) is issued • When slave mode is changed from receive mode to transmit mode
6	TEND	0	R/W	Transmit End [Clearing conditions] • When 0 is written in TEND after reading TEND = 1 • When data is written to ICDRT [Setting conditions] • When the ninth clock of SCL rises with the I²C bus format while the TDRE flag is 1 • When the final bit of transmit frame is sent with the clocked synchronous serial format

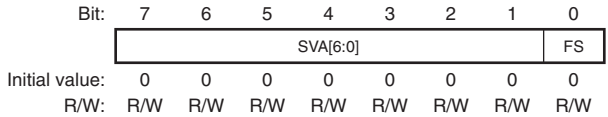
Bit	Bit Name	Initial Value	R/W	Description
5	RDRF	0	R/W	Receive Data Full [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When ICDRR is read [Setting condition] • When a receive data is transferred from ICDRS to ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Clearing condition] • When 0 is written in NACKF after reading NACKF = 1 [Setting condition] • When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1
3	STOP	0	R/W	Stop Condition Detection Flag [Clearing condition] • When 0 is written in STOP after reading STOP = 1 [Setting conditions] • In master mode, when a stop condition is detected after frame transfer • In slave mode, when the slave address in the first byte after the general call and detecting start condition matches the address set in SAR, and then the stop condition is detected

Bit	Bit Name	Initial Value	R/W	Description
2	AL/OVE	0	R/W	Arbitration Lost Flag/Overrun Error Flag Indicates that arbitration was lost in master mode with the I²C bus format and that the final bit has been received while RDRF = 1 with the clocked synchronous format. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface 3 detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been occupied by another master. [Clearing condition] • When 0 is written in AL/OVE after reading AL/OVE = 1 [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • When the SDA pin outputs high in master mode while a start condition is detected • When the final bit is received with the clocked synchronous format while RDRF = 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA[6:0] in SAR. [Clearing condition] • When 0 is written in AAS after reading AAS = 1 [Setting conditions] • When the slave address is detected in slave receive mode • When the general call address is detected in slave receive mode.
0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode with the I²C bus format. [Clearing condition] • When 0 is written in ADZ after reading ADZ = 1 [Setting condition] • When the general call address is detected in slave receive mode

16.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that selects the communications format and sets the slave address. In slave mode with the I²C bus format, if the upper seven bits of SAR match the upper seven bits of the first frame received after a start condition, this module operates as the slave device.

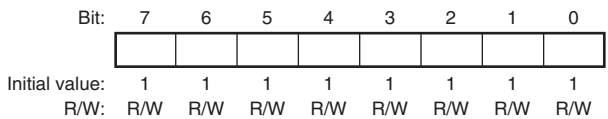
SAR is initialized to H'00 by a power-on reset.



Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA[6:0]	0000000	R/W	Slave Address These bits set a unique address in these bits, differing form the addresses of other slave devices connected to the I²C bus.
0	FS	0	R/W	Format Select 0: I²C bus format is selected 1: Clocked synchronous serial format is selected

16.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. ICDRT is initialized to H'FF.



16.3.8 I²C Bus Receive Data Register (ICDRR)

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the receive data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot write to this register.

ICDRR is initialized to H'FF by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.3.9 I²C Bus Shift Register (ICDRS)

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read directly from the CPU.

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

16.3.10 NF2CYC Register (NF2CYC)

NF2CYC is an 8-bit readable/writable register that selects the range of the noise filtering for the SCL and SDA pins. For details of the noise filter, see section 16.4.7, Noise Filter.

NF2CYC is initialized to H'00 by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	NF2CYC
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	NF2CYC	0	R/W	Noise Filtering Range Select 0: The noise less than one cycle of the peripheral clock can be filtered out 1: The noise less than two cycles of the peripheral clock can be filtered out

16.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

16.4.1 I²C Bus Format

Figure 16.3 shows the I²C bus formats. Figure 16.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

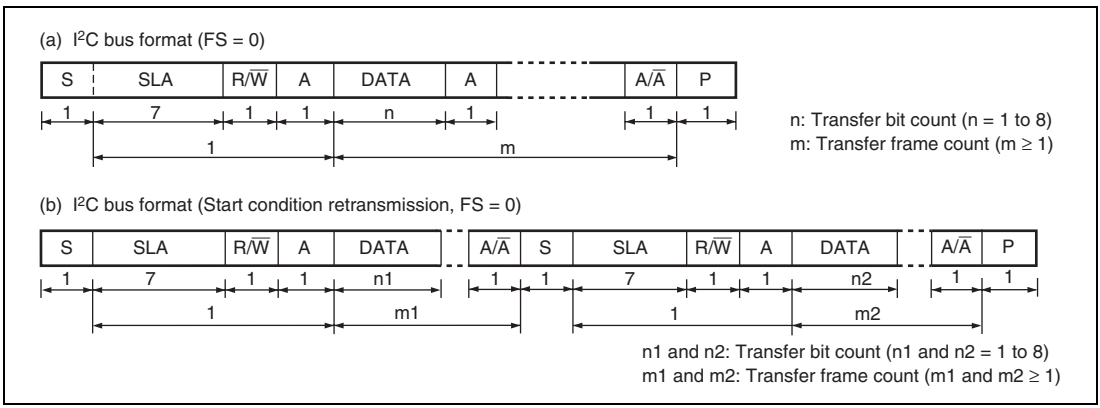


Figure 16.3 I²C Bus Formats

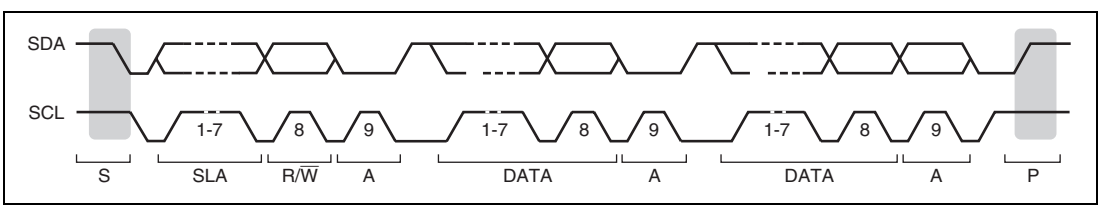


Figure 16.4 I²C Bus Timing

[Legend]

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/W: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives SDA to low.

DATA: Transfer data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

16.4.2 Master Transmit Operation

In master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 16.5 and 16.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Also, set the bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to slave receive mode.

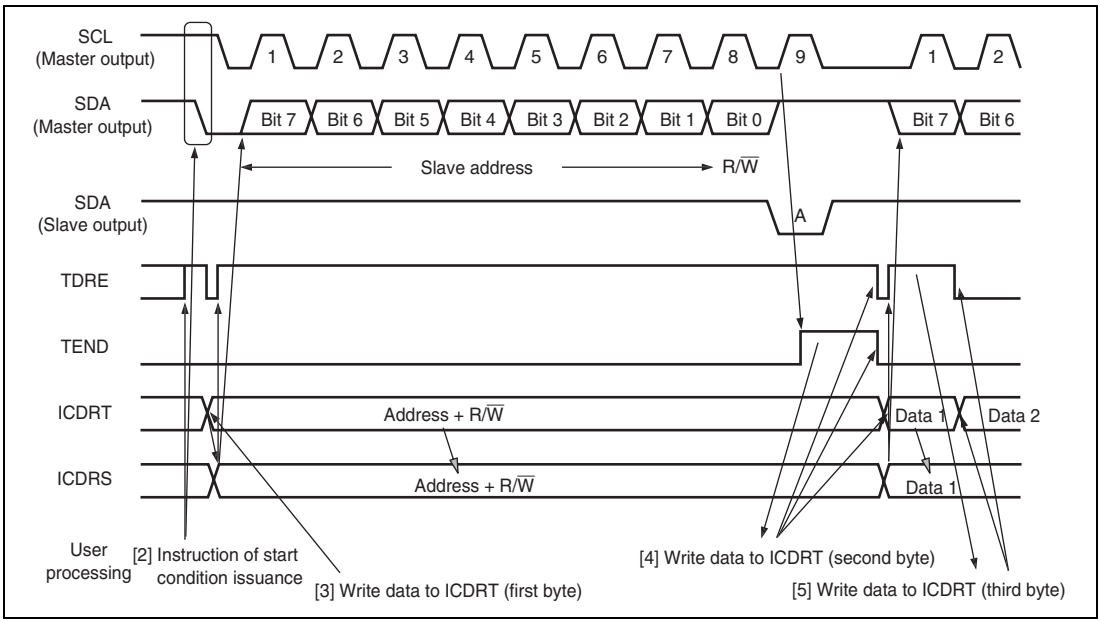


Figure 16.5 Master Transmit Mode Operation Timing (1)

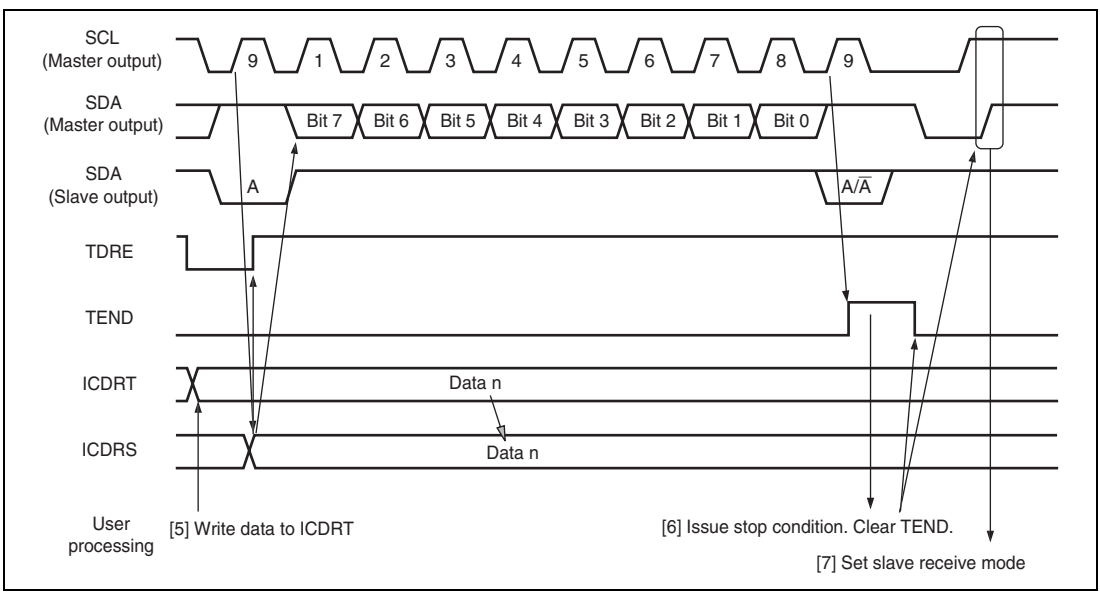


Figure 16.6 Master Transmit Mode Operation Timing (2)

16.4.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 16.7 and 16.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCR1 to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICSR is set to 1 at the rise of 9th receive clock pulse. At this time, the receive data is read by reading ICDRR, and RDRF is cleared to 0.
4. The continuous reception is performed by reading ICDRR every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, issue the stage condition.
7. When the STOP bit in ICSR is set to 1, read ICDRR. Then clear the RCVD bit to 0.
8. The operation returns to slave receive mode.

Note: If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

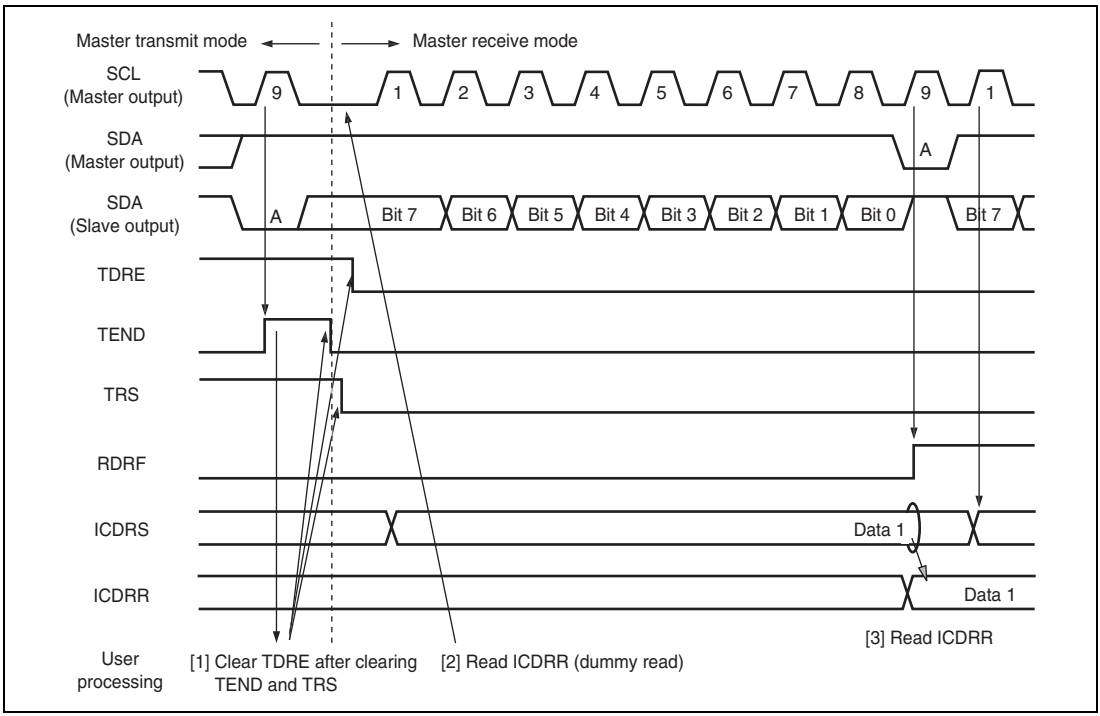


Figure 16.7 Master Receive Mode Operation Timing (1)

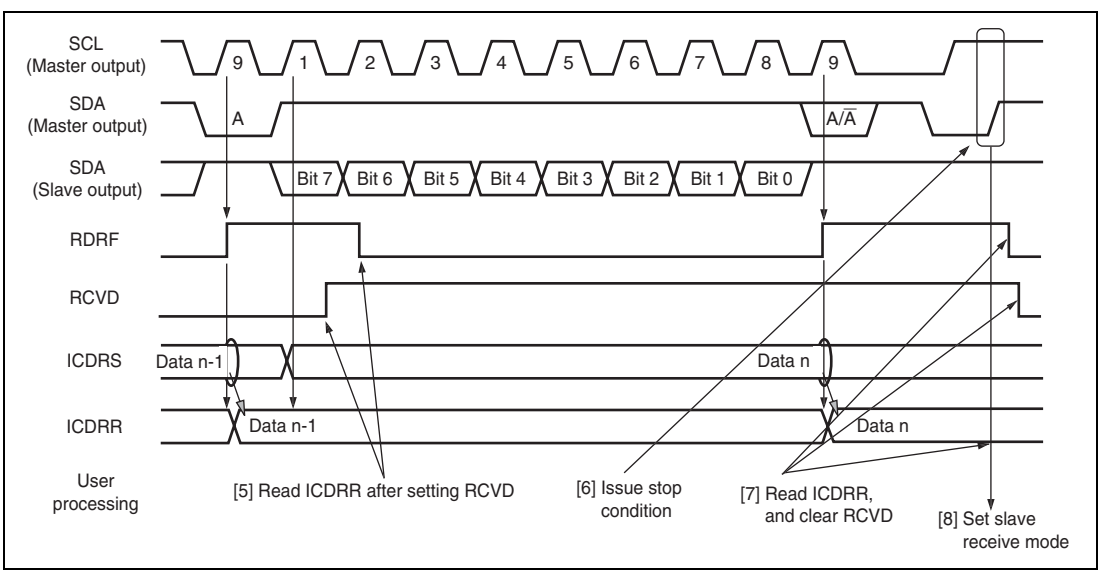


Figure 16.8 Master Receive Mode Operation Timing (2)

16.4.4 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 16.9 and 16.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data (R/$\overline{W}$) is 1, the TRS bit in ICCR1 and the TDRE bit in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is opened.
5. Clear TDRE.

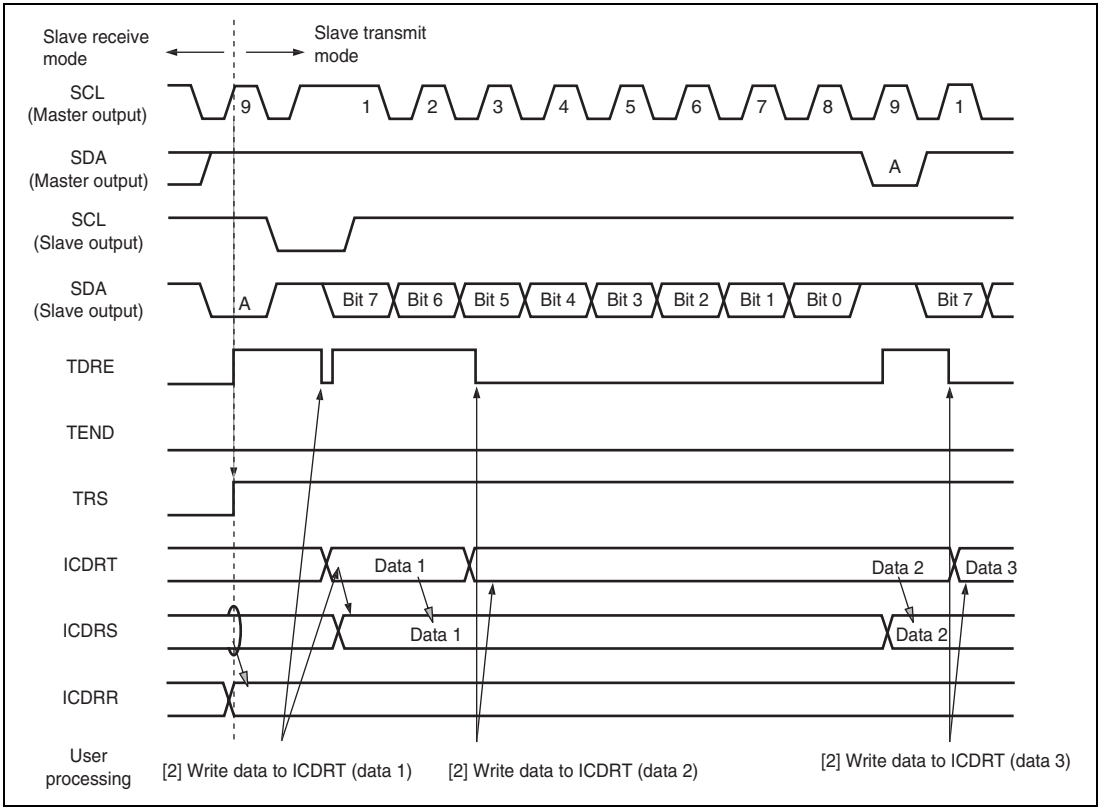


Figure 16.9 Slave Transmit Mode Operation Timing (1)

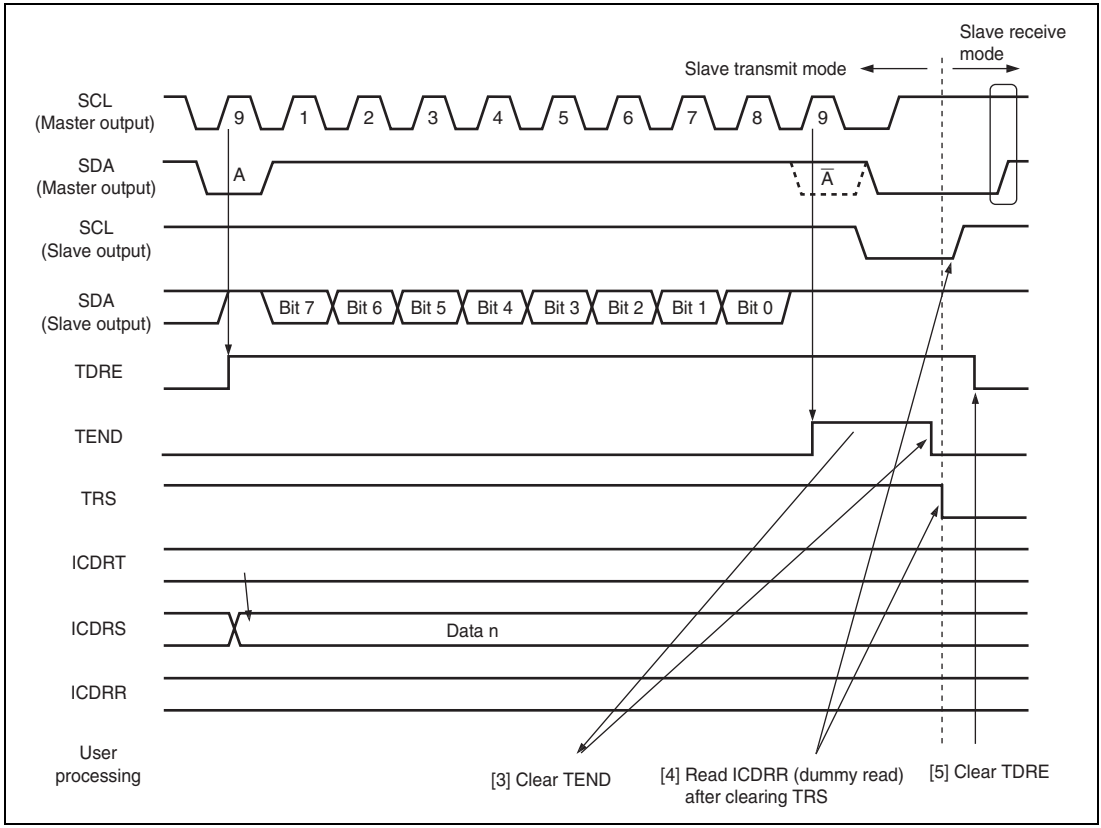


Figure 16.10 Slave Transmit Mode Operation Timing (2)

16.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 16.11 and 16.12. The reception procedure and operations in slave receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting) Set the MST and TRS bits in ICCR1 to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read). (Since the read data show the slave address and R/$\overline{W}$, it is not used.)
3. Read ICDRR every time RDRF is set. If 8th receive clock pulse falls while RDRF is 1, SCL is fixed low until ICDRR is read. The change of the acknowledge before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.
4. The last byte data is read by reading ICDRR.

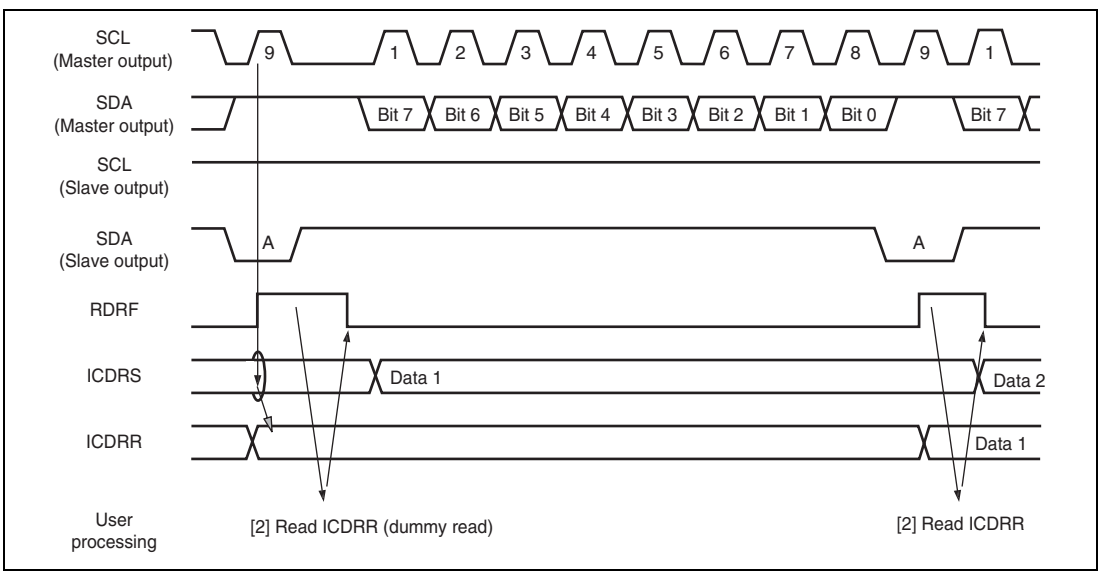


Figure 16.11 Slave Receive Mode Operation Timing (1)

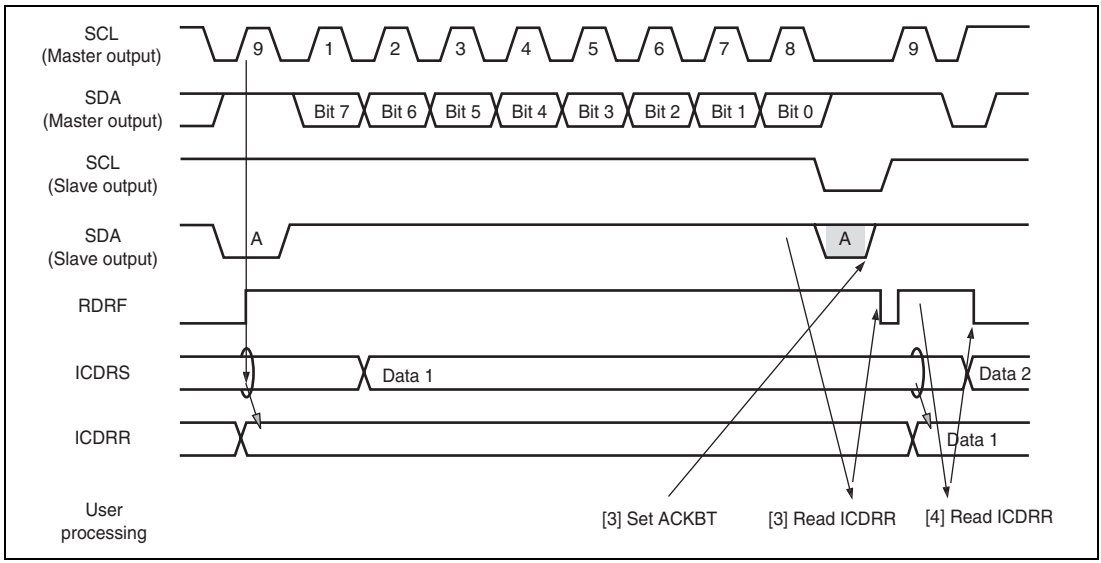


Figure 16.12 Slave Receive Mode Operation Timing (2)

16.4.6 Clocked Synchronous Serial Format

This module can be operated with the clocked synchronous serial format, by setting the FS bit in SAR to 1. When the MST bit in ICCR1 is 1, the transfer clock output from SCL is selected. When MST is 0, the external clock input is selected.

(1) Data Transfer Format

Figure 16.13 shows the clocked synchronous serial transfer format.

The transfer data is output from the fall to the fall of the SCL clock, and the data at the rising edge of the SCL clock is guaranteed. The MLS bit in ICMR sets the order of data transfer, in either the MSB first or LSB first. The output level of SDA can be changed during the transfer wait, by the SDAO bit in ICCR2.

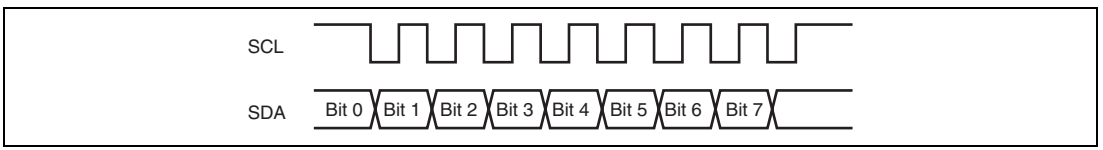


Figure 16.13 Clocked Synchronous Serial Transfer Format

(2) Transmit Operation

In transmit mode, transmit data is output from SDA, in synchronization with the fall of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For transmit mode operation timing, refer to figure 16.14. The transmission procedure and operations in transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the MST and CKS[3:0] bits in ICCR1. (Initial setting)
2. Set the TRS bit in ICCR1 to select transmit mode. Then, TDRE in ICSR is set.
3. Confirm that TDRE has been set. Then, write the transmit data to ICDRT. The data is transferred from ICDRT to ICDRS, and TDRE is set automatically. The continuous transmission is performed by writing data to ICDRT every time TDRE is set. When changing from transmit mode to receive mode, clear TRS while TDRE is 1.

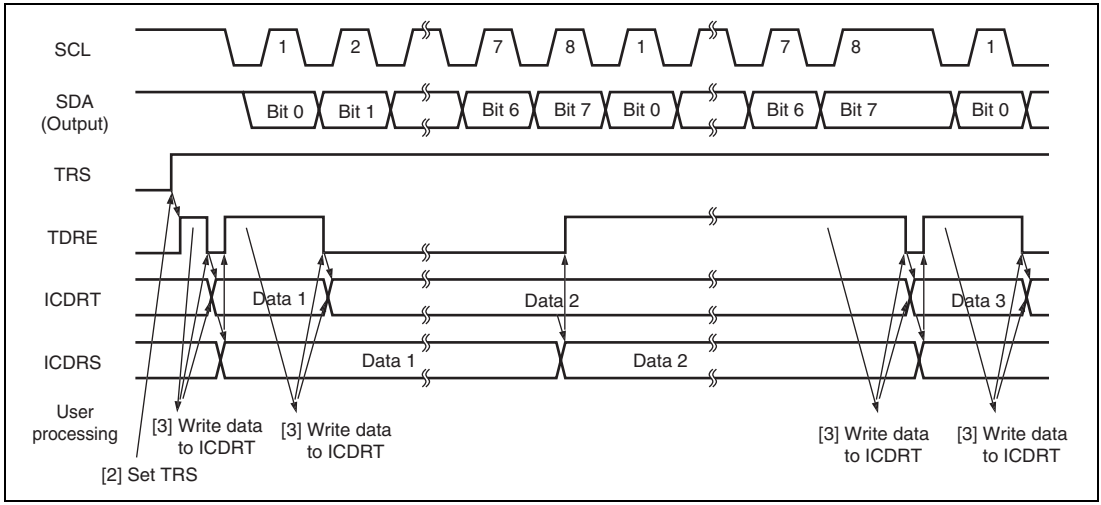


Figure 16.14 Transmit Mode Operation Timing

(3) Receive Operation

In receive mode, data is latched at the rise of the transfer clock. The transfer clock is output when MST in ICCR1 is 1, and is input when MST is 0. For receive mode operation timing, refer to figure 16.15. The reception procedure and operations in receive mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. When the transfer clock is output, set MST to 1 to start outputting the receive clock.
3. When the receive operation is completed, data is transferred from ICDRS to ICDRR and RDRF in ICSR is set. When MST = 1, the next byte can be received, so the clock is continually output. The continuous reception is performed by reading ICDRR every time RDRF is set. When the 8th clock is risen while RDRF is 1, the overrun is detected and AL/OVE in ICSR is set. At this time, the previous reception data is retained in ICDRR.
4. To stop receiving when MST = 1, set RCVD in ICCR1 to 1, then read ICDRR. Then, SCL is fixed high after receiving the next byte data.

Notes: Follow the steps below to receive only one byte with MST = 1 specified. See figure 16.16 for the operation timing.

1. Set the ICE bit in ICCR1 to 1. Set bits CKS[3:0] in ICCR1. (Initial setting)
2. Set MST = 1 while the RCVD bit in ICCR1 is 0. This causes the receive clock to be output.
3. Check if the BC2 bit in ICMR is set to 1 and then set the RCVD bit in ICCR1 to 1. This causes the SCL to be fixed to the high level after outputting one byte of the receive clock.

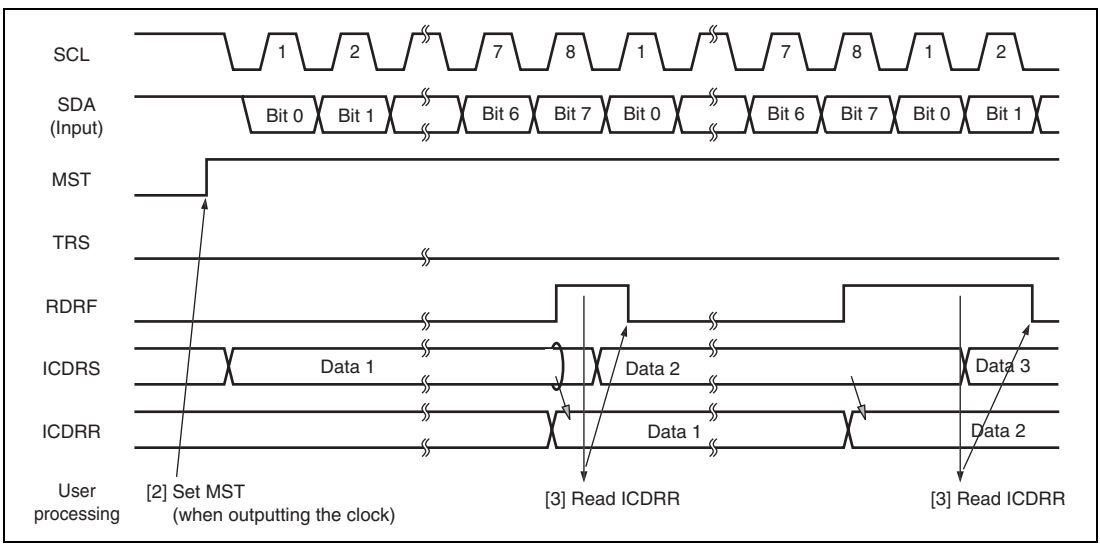


Figure 16.15 Receive Mode Operation Timing

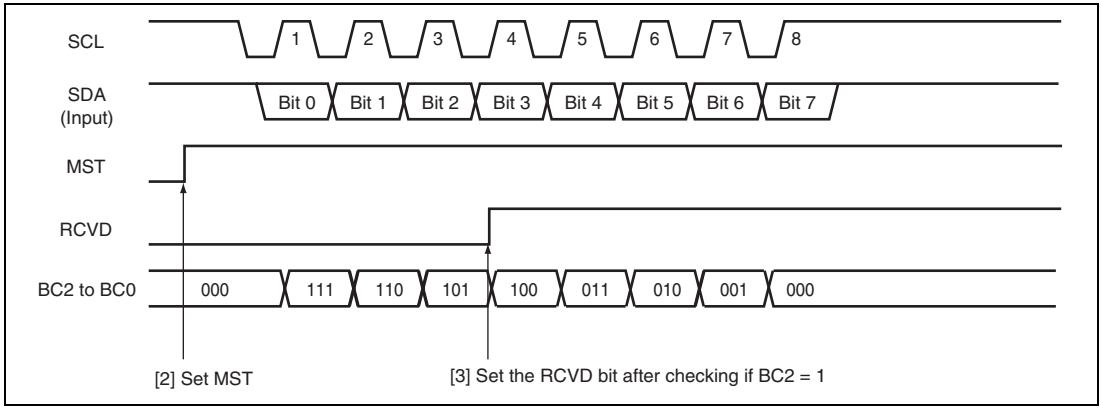


Figure 16.16 Operation Timing For Receiving One Byte (MST = 1)

16.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 16.17 shows a block diagram of the noise filter circuit.

The noise filter consists of three cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the peripheral clock. When NF2CYC is set to 0, this signal is not passed forward to the next circuit unless the outputs of both latches agree. When NF2CYC is set to 1, this signal is not passed forward to the next circuit unless the outputs of three latches agree. If they do not agree, the previous value is held.

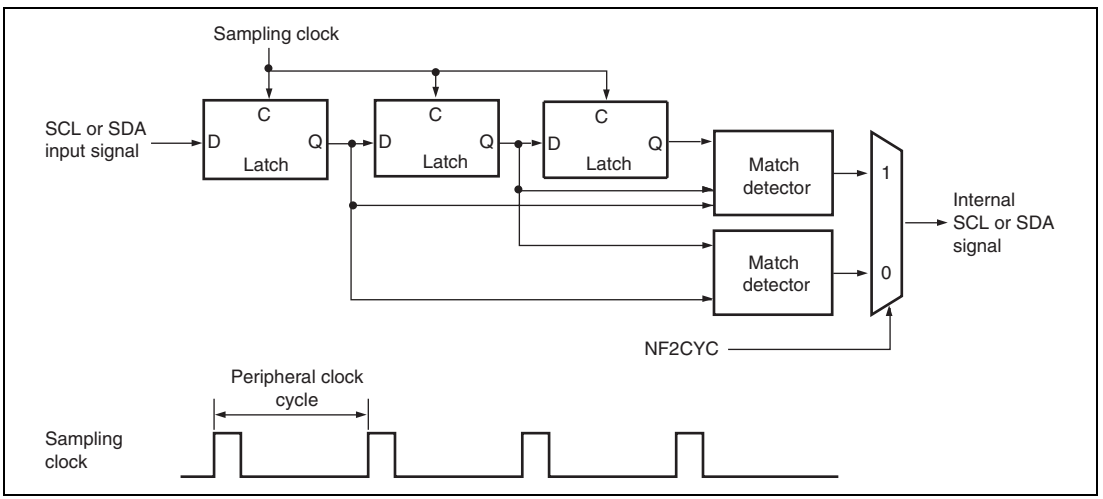


Figure 16.17 Block Diagram of Noise Filter

16.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 16.18 to 16.21.

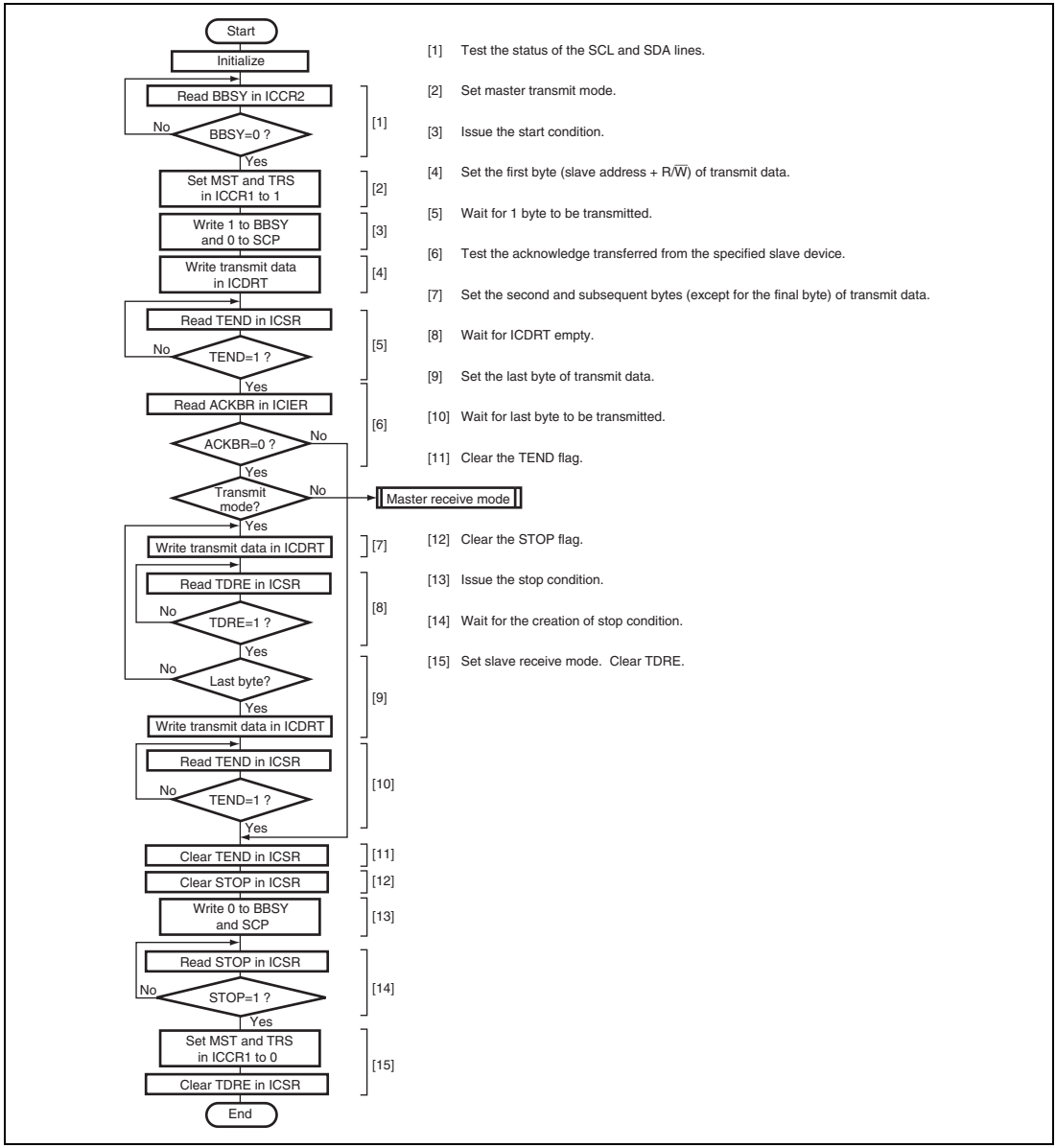


Figure 16.18 Sample Flowchart for Master Transmit Mode

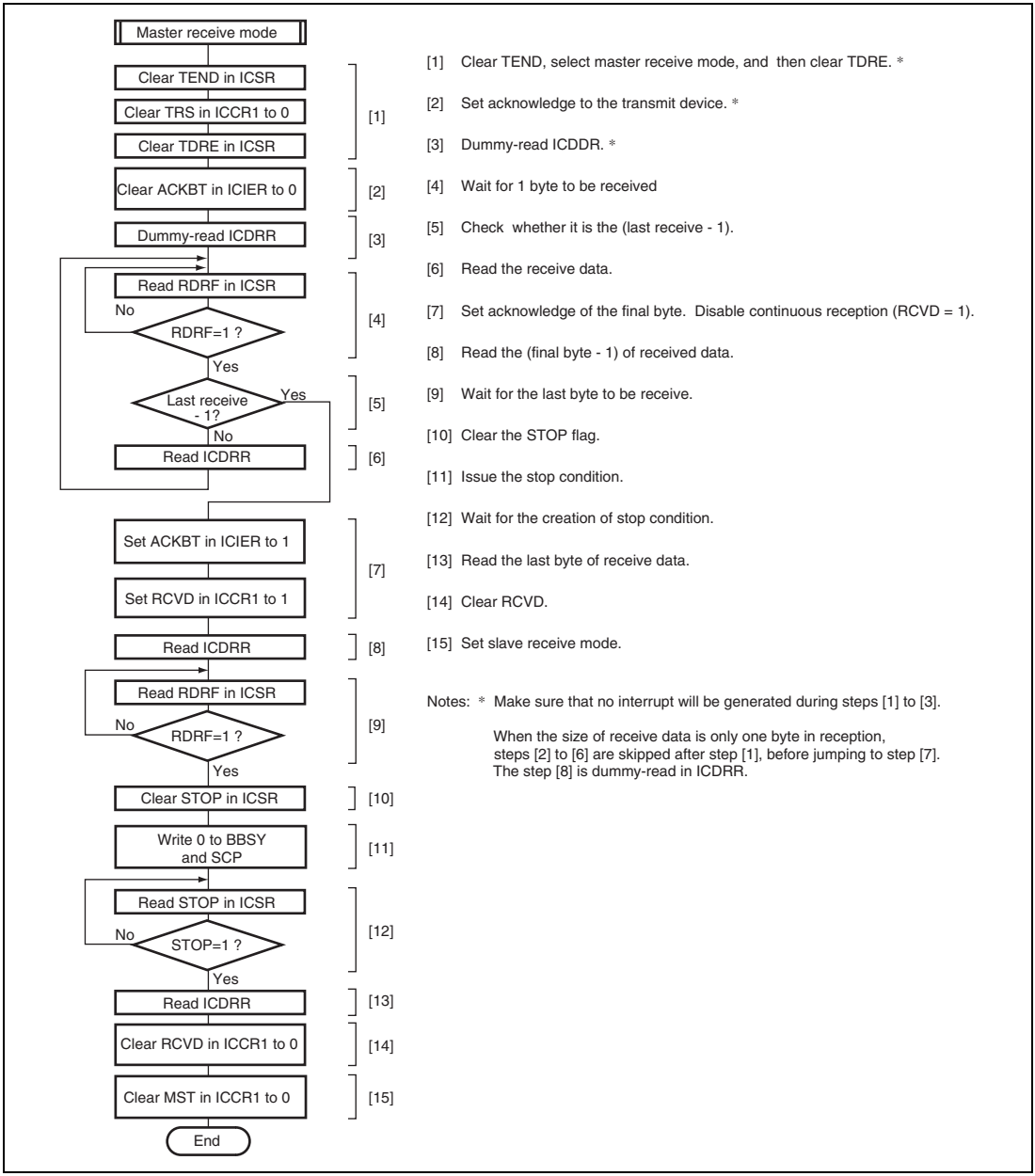


Figure 16.19 Sample Flowchart for Master Receive Mode

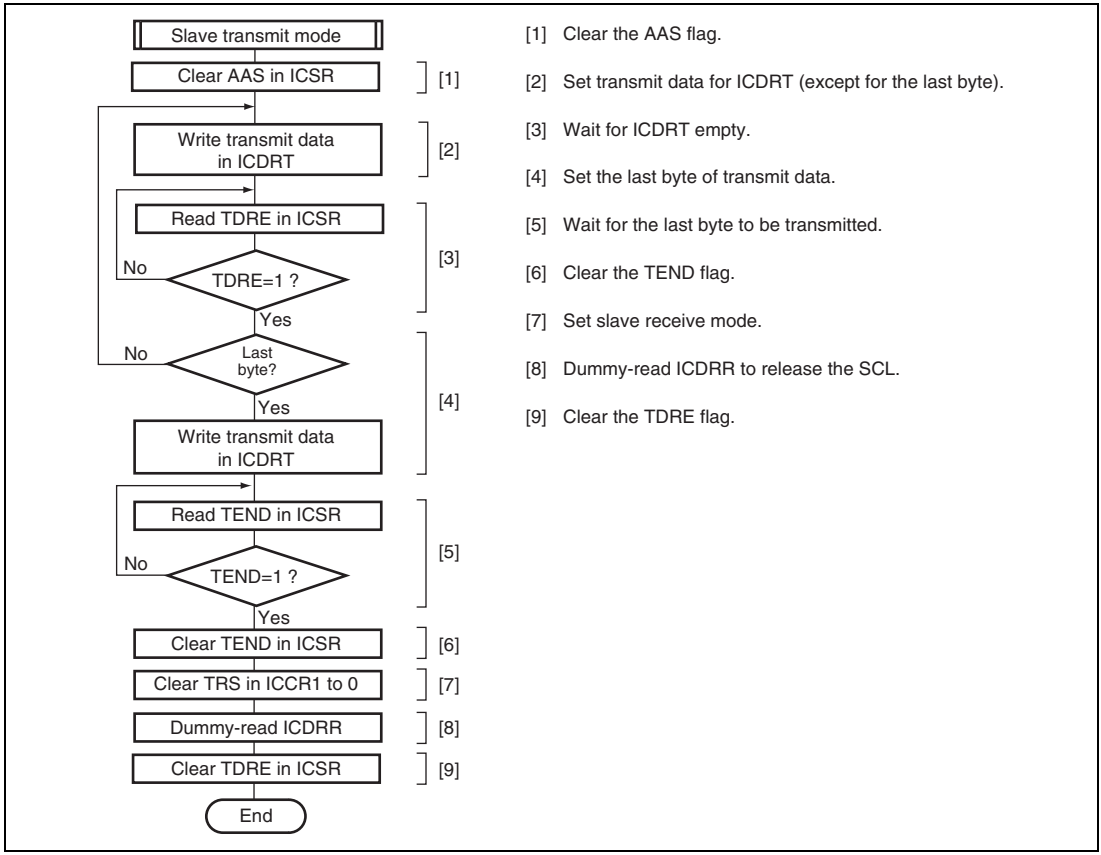


Figure 16.20 Sample Flowchart for Slave Transmit Mode

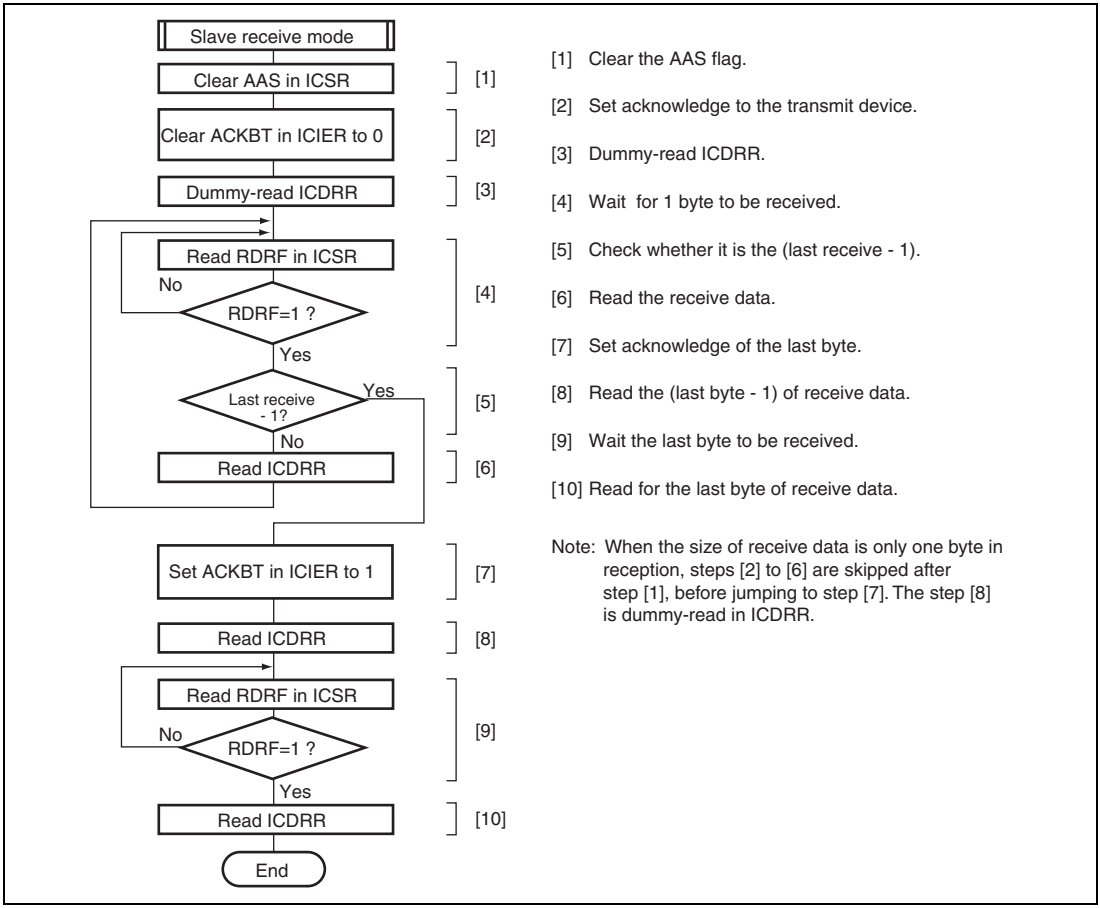


Figure 16.21 Sample Flowchart for Slave Receive Mode

16.5 Interrupt Requests

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK detection, STOP recognition, and arbitration lost/overrun error. Table 16.4 shows the contents of each interrupt request.

Table 16.4 Interrupt Requests

Interrupt Request	Abbreviation	Interrupt Condition	I²C Bus Format	Clocked Synchronous Serial Format
Transmit data Empty	TXI	(TDRE = 1) • (TIE = 1)	√	√
Transmit end	TEI	(TEND = 1) • (TEIE = 1)	√	√
Receive data full	RXI	(RDRF = 1) • (RIE = 1)	√	√
STOP recognition	STPI	(STOP = 1) • (STIE = 1)	√	—
NACK detection	NAKI	{(NACKF = 1) + (AL = 1)} • (NAKIE = 1)	√	—
Arbitration lost/ overrun error			√	√

When the interrupt condition described in table 16.4 is 1, the CPU executes an interrupt exception handling. Note that a TXI or RXI interrupt can activate the DMAC if the setting for DMAC activation has been made. In such a case, an interrupt request is not sent to the CPU. Interrupt sources should be cleared in the exception handling. The TDRE and TEND bits are automatically cleared to 0 by writing the transmit data to ICDRT. The RDRF bit is automatically cleared to 0 by reading ICDRR. The TDRE bit is set to 1 again at the same time when the transmit data is written to ICDRT. Therefore, when the TDRE bit is cleared to 0, then an excessive data of one byte may be transmitted.

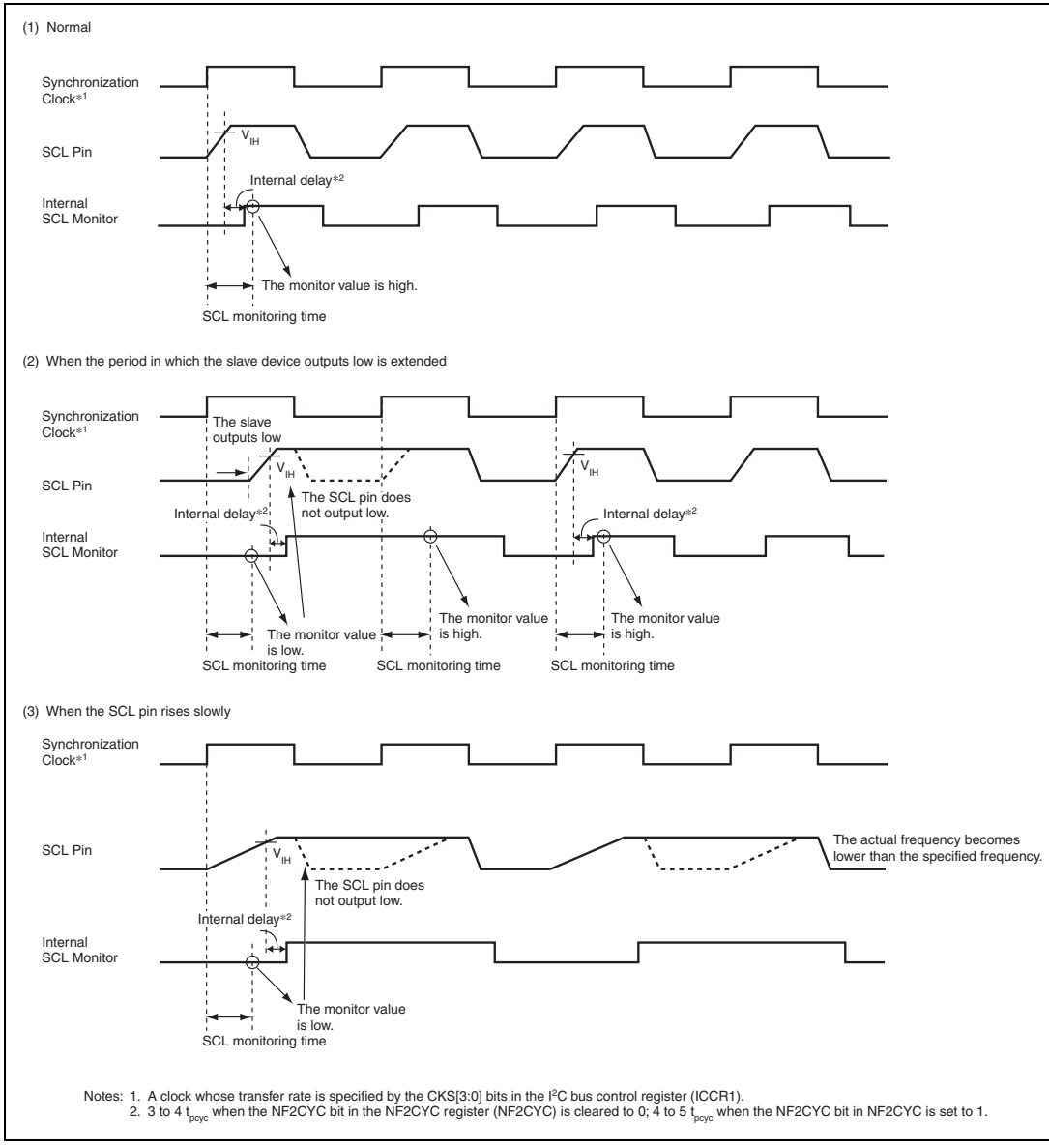
16.6 Bit Synchronous Circuit

In master mode, this module has a possibility that high level period may be short in the two states described below.

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lowered by the load of the SCL line (load capacitance or pull-up resistance)

Therefore, it monitors SCL and communicates by bit with synchronization.

Figure 16.22 shows the timing of the bit synchronous circuit and table 16.5 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.



Notes: 1. A clock whose transfer rate is specified by the CKS[3:0] bits in the I²C bus control register (ICCR1).
2. 3 to 4 t_{pcyc} when the NF2CYC bit in the NF2CYC register (NF2CYC) is cleared to 0; 4 to 5 t_{pcyc} when the NF2CYC bit in NF2CYC is set to 1.

Figure 16.22 Bit Synchronous Circuit Timing

Table 16.5 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL
0	0	9 tpcyc*
	1	21 tpcyc*
1	0	33 tpcyc*
	1	81 tpcyc*

Note: * tpcyc stands for the peripheral clock (Pϕ) cycle.

16.7 Usage Notes

16.7.1 Note on Multiple Master Usage

With multi-master used, if the transfer rate setting (CKS[3:0] in ICCR1) of I²C in this LSI is slower than the other masters, the SCL with unexpected width may be output in rare cases.

To prevent this problem, the transfer rate of I²C should be specified as equal to or higher than 1/1.8 of the highest transfer rate among the other masters.

16.7.2 Note on Master Receive Mode

If ICDRR is read near the falling edge of 8th clock, the receive data will not be received in some cases. In addition, if RCVD is set to 1 near the falling edge of 8th clock, a stop condition cannot be issued in some cases. To prevent these errors, one of the following two methods should be selected.

1. In master receive mode, ICDRR should be read before the falling edge of 8th clock.
2. In master receive mode, RCVD should be set to 1 and the processing should be performed in byte units.

16.7.3 Note on Master Receive Mode with ACKBT Setting

In master receive mode operation, ACKBT should be set before the 8th falling edge of SCL in the final data transfer during continuous data transfer. Otherwise, the slave device may overrun.

16.7.4 Note on MST and TRS Bit Status When an Arbitration was Lost

If the master transmission is set according to the MST and TRS bit settings while multiple masters are used, the conflicting status in which the AL bit in ICSR is set to 1 in master transmit mode (MST and TRS are set to 1) depending on the arbitration lost generation timing during TRS bit handling instruction execution.

This problem can be avoided by the following methods.

- When multiple masters are used, the MST and TRS bits should be set by a MOV instruction.
- When an arbitration lost occurs, check if both MST and TRS bits are cleared to 0. If either or both of MST and TRS bits are not cleared to 0, both the bits should be cleared to 0.

Section 17 A/D Converter (ADC)

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

17.1 Features

- 12-bit resolution
- Input channels

  Eight channels (two independent A/D conversion modules)
- High-speed conversion

  When Aϕ = 40 MHz: Minimum 1.25 μs per channel

  AD clock = 40 MHz, 50 conversion states
- Two operating modes
  - Single-cycle scan mode: Continuous A/D conversion on one to eight channels
  - Continuous scan mode: Repetitive A/D conversion on one to eight channels
- A/D data registers

  Eight A/D data registers (ADDR) are provided. A/D conversion results are stored in A/D data registers (ADDR) that correspond to the input channels.
- Sample-and-hold function

  A sample-and-hold circuit is 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 for channels 0 to 2 and 8 to 10.
  - Group A (GrA): Analog input pins selected from channels 0, 1, and 2 can be simultaneously sampled.
- Three methods for starting conversion

  Software: Setting of the ADST bit in ADCR

  Timer: TRGAN, TRG0N, TRG4AN, and TRG4BN from the MTU2
  TRGAN, TRG4AN, and TRG4BN from the MTU2S

  External trigger: $\overline{\text{ADTRG}}$ (LSI pin)
- Selectable analog input channel

  A/D conversion of a selected channel is accomplished by setting the A/D analog input channel select registers (ADANSR).
- A/D conversion end interrupt and DMAC transfer function is supported

  On completion of A/D conversion, A/D conversion end interrupts (ADI) can be generated and the DMAC can be activated by ADI.

- Module standby mode can be set

Figure 17.1 shows a block diagram of the A/D converter.

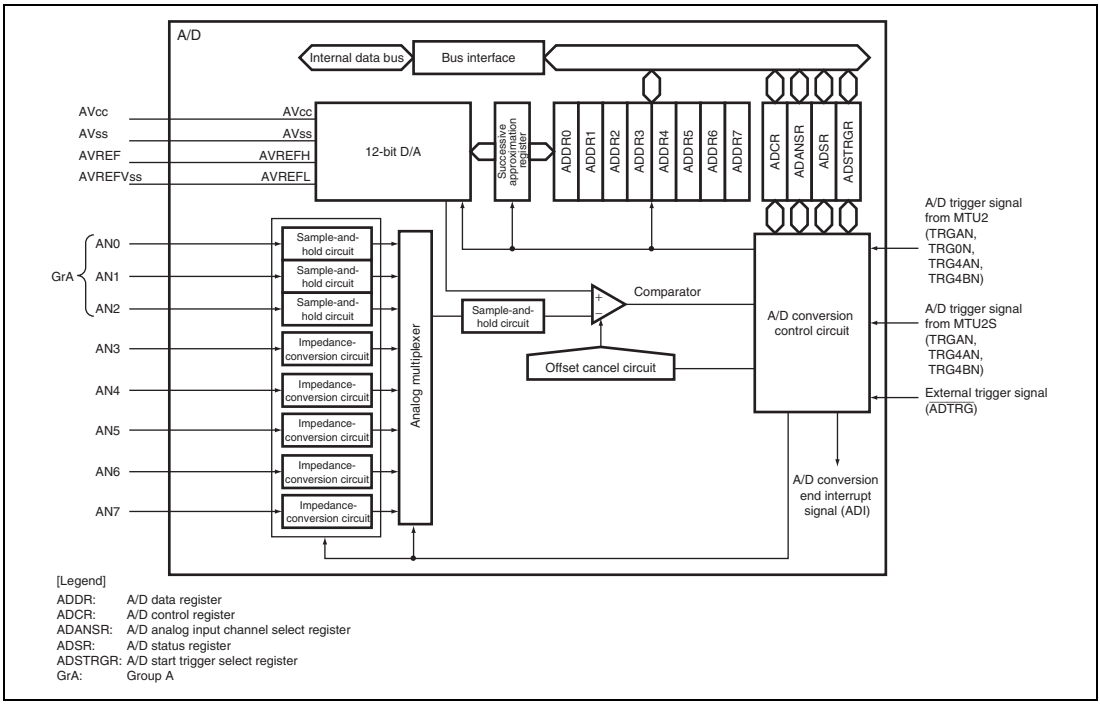


Figure 17.1 Block Diagram of A/D Converter

17.2 Input/Output Pins

Table 17.1 shows the configuration of the pins used by the A/D converter. For the pin usage, refer to the usage notes in section 17.7, Usage Notes.

Table 17.1 Pin Configuration

Pin Name	I/O	Function
AVcc	Input	Analog block power supply pin
AVss	Input	Analog block ground pin
AVREF	Input	Analog block reference power supply pin
AVREFVss	Input	Analog block reference ground pin
$\overline{\text{ADTRG}}$	Input	A/D external trigger input pin
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
AN4	Input	Analog input pin 4
AN5	Input	Analog input pin 5
AN6	Input	Analog input pin 6
AN7	Input	Analog input pin 7

17.3 Register Descriptions

The A/D converter has the following registers.

Table 17.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D control register	ADCR	R/W	H'00	H'FFFFE800	8
A/D status register	ADSR	R/W	H'00	H'FFFFE802	8
A/D start trigger select register	ADSTRGR	R/W	H'00	H'FFFFE81C	8
A/D analog input channel select register	ADANSR	R/W	H'00	H'FFFFE820	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 data register 4	ADDR4	R	H'0000	H'FFFFE848	16
A/D data register 5	ADDR5	R	H'0000	H'FFFFE84A	16
A/D data register 6	ADDR6	R	H'0000	H'FFFFE84C	16
A/D data register 7	ADDR7	R	H'0000	H'FFFFE84E	16

17.3.1 A/D Control Register (ADCR)

ADCR is an 8-bit readable/writable register that selects A/D conversion mode and others.

Bit:	7	6	5	4	3	2	1	0
	ADST	ADCS	ACE	ADIE	-	-	TRGE	EXTRG
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	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, 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, or module 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. When the external trigger is used as an A/D conversion start trigger, the low-level pulse input to the ADTRG pin must be at least 1.5 Aϕ 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 ($\overline{\text{ADTRG}}$)

17.3.2 A/D Status Register (ADSR)

ADSR is an 8-bit readable/writable register that indicates the status of the A/D converter.

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

Note: * Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way. Do not overwrite this bit with 0 when the value of this bit 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

17.3.3 A/D Start Trigger Select Register (ADSTRGR)

ADSTRGR selects an A/D conversion start trigger from the MTU2 or MTU2S. The A/D conversion start trigger is used as an A/D conversion start source when the TRGE bit in ADCR is set to 1 and the EXTRG bit in ADCR is set to 0.

Bit:	7	6	5	4	3	2	1	0
	-	STR6	STR5	STR4	STR3	STR2	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	STR6	0	R/W	Start Trigger 6 Enables or disables the A/D conversion start request input from the MTU2S. 0: Disables the A/D conversion start by TRGAN trigger (MTU2S). 1: Enables the A/D conversion start by TRGAN trigger (MTU2S).
5	STR5	0	R/W	Start Trigger 5 Enables or disables the A/D conversion start request input from the MTU2S. 0: Disables the A/D conversion start by TRG4AN trigger (MTU2S). 1: Enables the A/D conversion start by TRG4AN trigger (MTU2S).
4	STR4	0	R/W	Start Trigger 4 Enables or disables the A/D conversion start request input from the MTU2S. 0: Disables the A/D conversion start by TRG4BN trigger (MTU2S). 1: Enables the A/D conversion start by TRG4BN trigger (MTU2S).

Bit	Bit Name	Initial Value	R/W	Description
3	STR3	0	R/W	Start Trigger 3 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRG0N trigger (MTU2). 1: Enables the A/D conversion start by TRG0N trigger (MTU2).
2	STR2	0	R/W	Start Trigger 2 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRGAN trigger (MTU2). 1: Enables the A/D conversion start by TRGAN trigger (MTU2).
1	STR1	0	R/W	Start Trigger 1 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRG4AN trigger (MTU2). 1: Enables the A/D conversion start by TRG4AN trigger (MTU2).
0	STR0	0	R/W	Start Trigger 0 Enables or disables the A/D conversion start request input from the MTU2. 0: Disables the A/D conversion start by TRG4BN trigger (MTU2). 1: Enables the A/D conversion start by TRG4BN trigger (MTU2).

17.3.4 A/D Analog Input Channel Select Register (ADANSR)

ADANSR is an 8-bit readable/writable register that selects 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 17.3. When changing the analog input channel, the ADST bit in ADCR must be cleared to 0 to prevent incorrect operations.
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 17.3 Channel Select List

Bit Name	Analog Input Channels
ANS0	AN0
ANS1	AN1
ANS2	AN2
ANS3	AN3
ANS4	AN4
ANS5	AN5
ANS6	AN6
ANS7	AN7

17.3.5 A/D Data Registers 0 to 7 (ADDR0 to ADDR7)

ADDRs are 16-bit read-only registers. The conversion result for each analog input channel is stored in ADDR with the corresponding number. (See table 17.4.)

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 17.4 Correspondence between Analog Channels and Registers (ADDR0 to ADDR7)

Analog Input Channels	A/D Data Registers
AN0	ADDR0
AN1	ADDR1
AN2	ADDR2
AN3	ADDR3
AN4	ADDR4
AN5	ADDR5
AN6	ADDR6
AN7	ADDR7

17.4 Operation

The A/D converter has two operating modes: single-cycle scan mode and continuous scan mode. In single-cycle scan mode, A/D conversion is performed once on one or more specified channels and then it ends. In continuous scan mode, the A/D conversion is performed sequentially on one or more specified channels until the ADST bit is cleared to 0.

The ADCS bit in the A/D control register (ADCR) is used to select the operating mode. Setting the ADCS bit to 0 selects single-cycle scan mode and setting the ADCS bit to 1 selects continuous scan mode. In both modes, A/D conversion starts on the channel with the lowest number in the analog input channels selected by the A/D analog input channel select register (ADANSR) from AN0 to AN7.

In single-cycle scan mode, when one cycle of A/D conversion on all specified channels is completed, the ADF bit in ADSR is set to 1 and the ADST bit is automatically cleared to 0. In continuous scan mode, when conversion on all specified channels is completed, the ADF bit in ADSR is set to 1. To stop A/D conversion, write 0 to the ADST bit. When the ADF bit is set to 1, if the ADIE bit in ADCR is set to 1, an A/D conversion end interrupt (ADI) is generated. When clearing the ADF bit to 0, read the ADF bit while set to 1 and then write 0. However, when the DMAC is activated by an ADI interrupt, the ADF bit is automatically cleared to 0.

17.4.1 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 ADST bit in the A/D control register (ADCR) to 1 to start A/D conversion.
4. After channels 0 to 2 (GrA) are sampled simultaneously, offset canceling processing (OFC) is performed. 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. Following this, channel 1 is converted. Upon completion of the conversion, the A/D conversion result is transferred to ADDR1. In the same way, channel 2 is converted and the A/D conversion result is transferred to ADDR2.

   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.

5. 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.

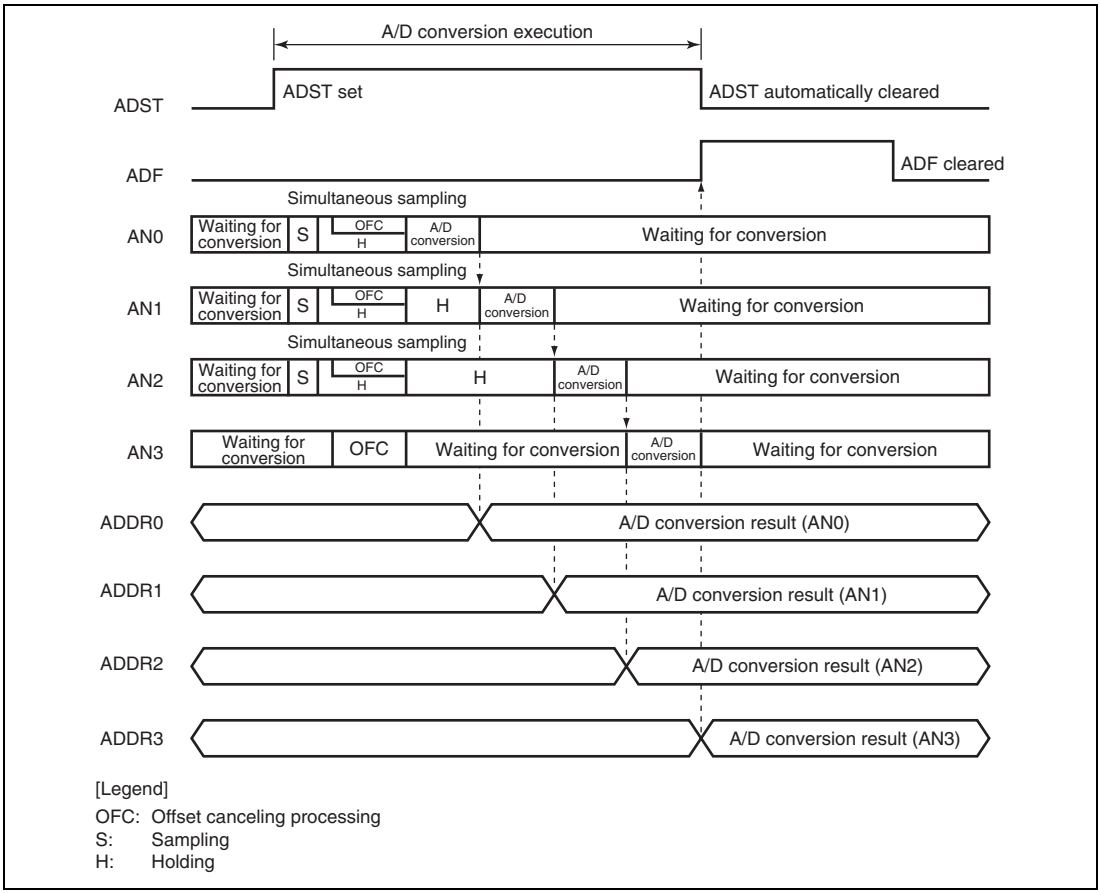


Figure 17.2 Example of A/D_0 Converter Operation (Single-Cycle Scan Mode)

17.4.2 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 ADST bit in the A/D control register (ADCR) to 1 to start A/D conversion.
4. Channels 0 and 2 (GrA) are sampled simultaneously. As the ANS1 bit in ADANSR is set to 0, channel 1 is not sampled. After this, offset canceling processing (OFC) is performed. 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.
5. The A/D conversion of channel 3 starts. Upon completion of the A/D conversion, the A/D conversion result is transferred to ADDR3.
6. When the A/D conversion ends on all the specified channels (AN0 to 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.
7. Steps 4 to 6 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 4 to 6.

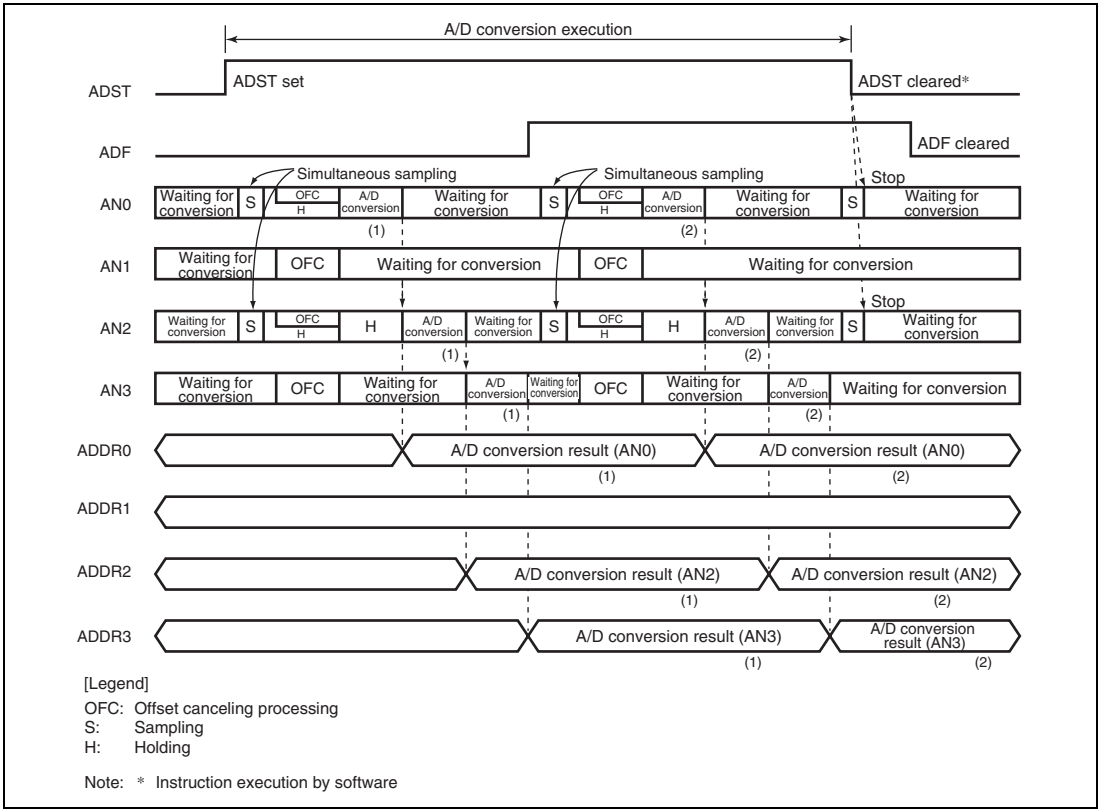


Figure 17.3 Example of A/D Converter Operation (Continuous Scan Mode)

17.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit common to all the channels. Each of channels 0 to 2 of the A/D converter has a dedicated built-in sample-and-hold circuit. Channels 0 to 2 can be simultaneously sampled as one group. This group is referred to as Group A (GrA) (in table 17.5). Even when only one channel is selected in the group by ADANSR, the sample-and-hold operation is performed with the dedicated sample-and-hold circuit. When only the channels without a dedicated sample-and-hold circuit are specified by ADANSR, the time that elapses is the same as when a dedicated sample-and-hold circuit is used.

When an event that sets the ADST bit, for example, writing to this bit by the CPU, A/D converter activation request from the MTU2, the MTU2S, or an external trigger signal occurs, the analog input is sampled by the dedicated sample-and-hold circuit for each channel after the A/D conversion start delay time (t_D) has passed and the offset canceling processing (OFC) is performed. After this, the sampling of the analog input using the sample-and-hold circuit common to all the channels is performed and then the A/D conversion is started. Figure 17.4 shows the A/D conversion timing in this case. This A/D conversion time (t_{CONV}) includes the t_D, the offset canceling processing time (t_{OFC}), the analog input sampling time with a dedicated sample-and-hold circuit for each channel (t_{SPLSH}), and the analog input sampling time with the sample-and-hold circuit common to all the channels (t_{SPL}). The t_{SPLSH} does not depend on the number of channels simultaneously sampled.

In continuous scan mode, the A/D conversion time (t_{CONV}) given in table 17.6 applies to the conversion time of the first cycle. The conversion time of the second and subsequent cycles is expressed as ($t_{CONV} - t_D + 6$).

Table 17.5 Correspondence between Analog Input Channels and Groups being Allowed Simultaneous Sampling

Analog Input Channels	Group
AN0	GrA
AN1	
AN2	
AN3	—
AN4	—
AN5	—
AN6	—
AN7	—

Table 17.6 A/D Conversion Time

Item	Symbol	Number of Required States		
		Min.	Typ.	Max.
A/D conversion start delay time	t_D	11*[1]	—	15*[2]
Analog input sampling time of dedicated sample-and-hold circuit for GrA and GrB	t_{SPLSH}	—	30	—
Offset canceling processing time	t_{OFC}	—	50	—
Analog input sampling time of sample-and-hold circuit common to all channels	t_{SPL}	—	20	—
A/D conversion complete processing	t_{end}	—	4	—
A/D conversion time	t_{CONV}	50n + 95*[3]	—	50n + 99*[3]

Notes: 1. A/D converter activation by the MTU2 or MTU2S trigger signal.
2. A/D converter activation by an external trigger signal.
3. n: number of A/D conversion channels (n = 1 to 8)

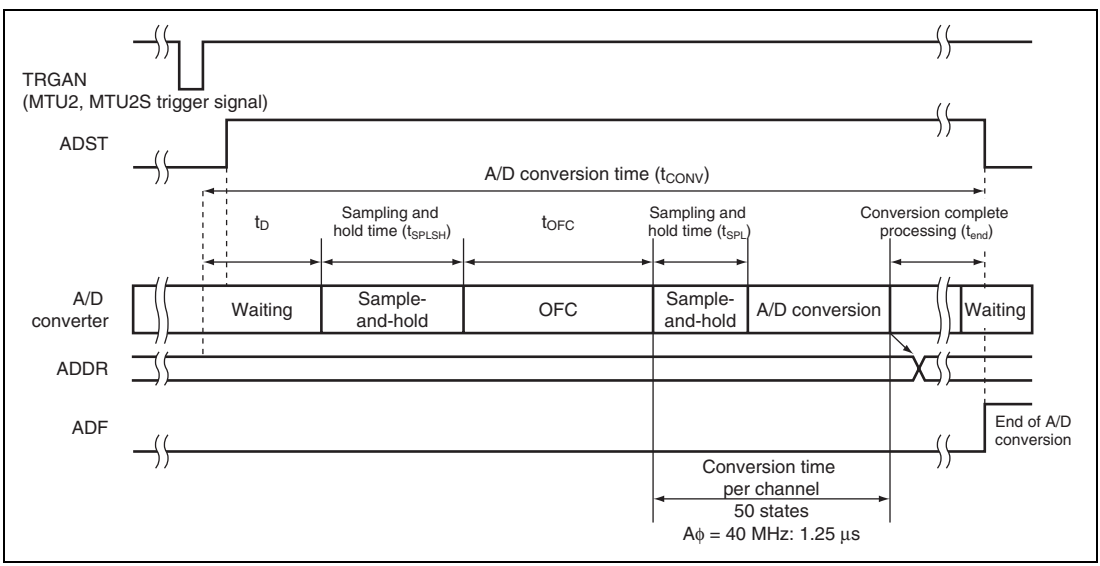


Figure 17.4 A/D Conversion Timing (Single-Cycle Scan Mode)

17.4.4 A/D Converter Activation by MTU2 and MTU2S

A/D conversion is activated by the A/D conversion start triggers (TRGAN, TRG0N, TRG4N, and TRG4BN) from the MTU2 and A/D conversion start triggers (TRGAN, TRG4AN, and TRG4BN) from the MTU2S. To enable this function, set the TRGE bit in ADCR to 1 and clear the EXTRG bit to 0. After this setting is made, if an A/D conversion start trigger from the MTU2 or MTU2S is generated, the ADST bit is set to 1. The timing between the setting of the ADST bit and the start of the A/D conversion is the same for all A/D conversion activation sources.

The A/D conversion start trigger must be input after ADCR, ADSTRGR, and ADANSR registers have been set.

17.4.5 External Trigger Input Timing

The A/D conversion can be externally triggered. To input an external trigger, set the pin function controller (PFC) to select $\overline{\text{ADTRG}}$ pin function and drive the $\overline{\text{ADTRG}}$ pin low when a high level is input to the $\overline{\text{ADTRG}}$ pin with the TRGE and EXTRG bits in ADCR are both set to 1. A falling edge of the $\overline{\text{ADTRG}}$ pin sets the ADST bit in ADCR to 1, starting the A/D conversion. Other operations are conducted in the same way for all A/D conversion activation sources. Figure 17.5 shows the timing.

The ADST bit is set to 1 after 5 states has elapsed from the point at which the A/D converter detects a falling edge on the $\overline{\text{ADTRG}}$ pin. A low level input to the $\overline{\text{ADTRG}}$ pin must be made after the ADCR, ADSTRGR, and ADANSR registers have been set.

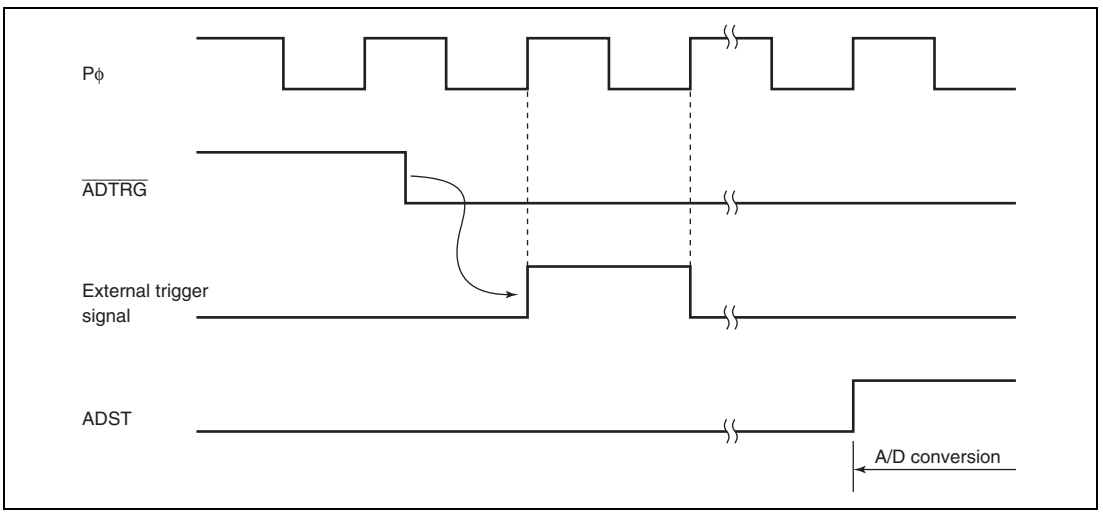


Figure 17.5 External Trigger Input Timing

17.4.6 Example of ADDR Auto-Clear Function

When the A/D data register (ADDR) is read by the CPU or DMAC, ADDR can be automatically cleared to H'0000 by setting the ACE bit in ADCR to 1. This function allows the detection of non-updated ADDR states.

Figure 17.6 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 or DMAC 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.

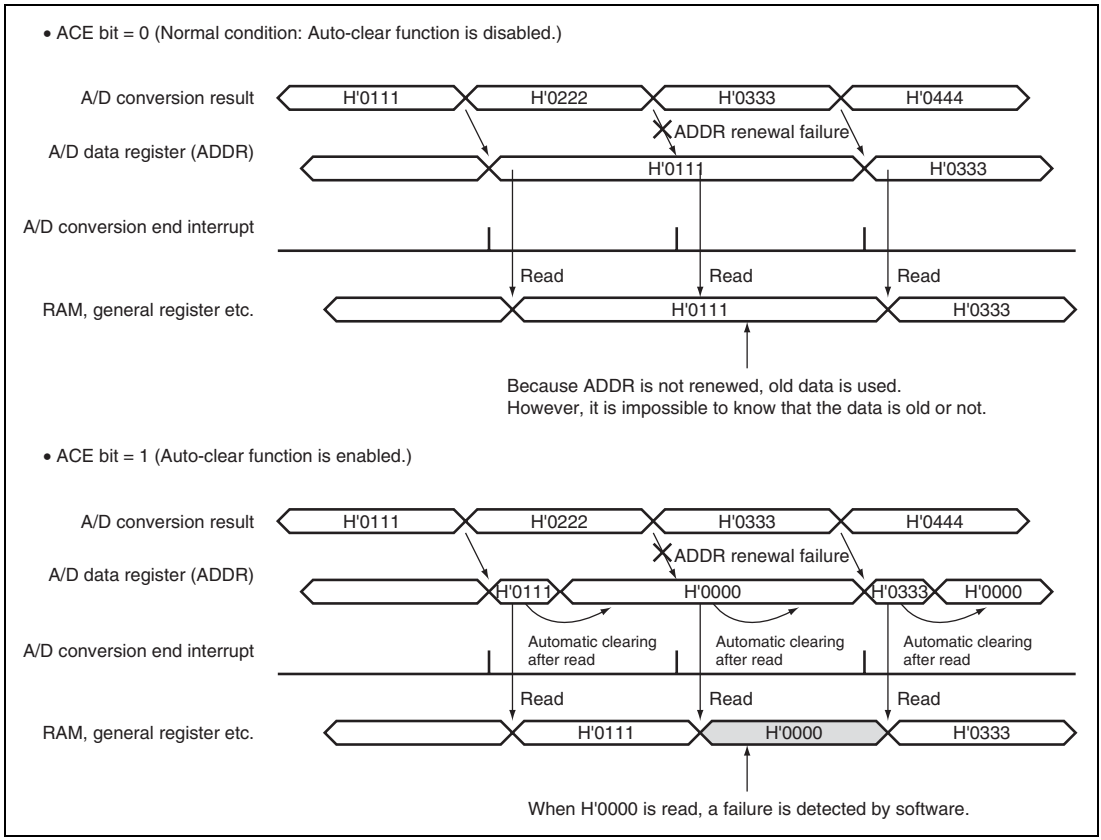


Figure 17.6 Example of When ADDR Auto-clear Function is Disabled (Normal Condition)/Enabled

17.5 Interrupt Sources and DMAC 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 can be activated by the DMAC setting when an ADI interrupt is generated. At this time, no interrupt to the CPU is generated. When the DMAC is activated by an ADI interrupt, the ADF bit in ADSR is automatically cleared at the data transfer by the DMAC.

17.6 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 17.7).
- 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 17.7).
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.7).
- 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 17.7).
- Absolute accuracy

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

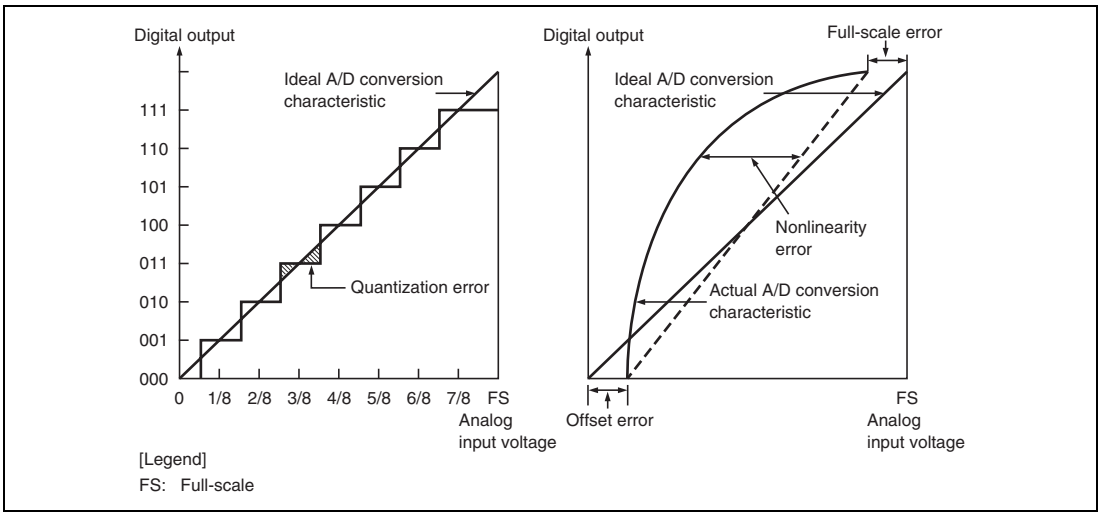


Figure 17.7 Definitions of A/D Conversion Accuracy

17.7 Usage Notes

17.7.1 Analog Input Voltage Range

The voltage applied to analog input pin (ANn) during A/D conversion should be in the range AVss ≤ ANn (n = 0 to 7) ≤ AVcc.

17.7.2 Relationship between AVcc, AVss and Vcc, Vss

When using the A/D converter, set AVcc = 5.0 V ±0.5 V and AVss = Vss. When the A/D converter is not used, set Vcc ≤ AVcc ≤ 5.0V ±0.5V, AVss = Vss, and do not leave the AVcc pin open.

17.7.3 Range of AVREF Pin Settings

Set AVREF = AVcc ±0.3 V and AVREFVss = AVss ±0.3 V. If these conditions are not met, the reliability of the LSI may be adversely affected.

17.7.4 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.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference power supply (AVREF), the analog power supply (AVcc), the analog ground (AVss), and the analog reference ground (AVREFVss). Also, AVss should be connected at one point to a stable digital ground (Vss) on the board.

17.7.5 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN7) and analog reference power supply (AVREF), a protection circuit should be connected between the AVcc and AVss, as shown in figure 17.8. The AVREFVss and AVss should be the same voltage. Also, the bypass capacitors connected to AVREF and the filter capacitor connected to ANn should be connected to the AVss. If a filter capacitor is connected as shown in figure 17.8, 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.

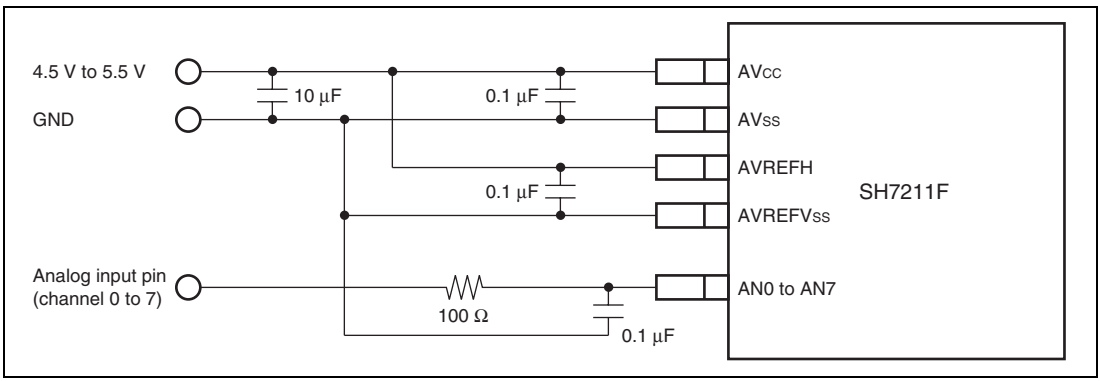


Figure 17.8 Example of Analog Input Pin Protection Circuit

17.7.6 Notes on Register Setting

- Set the ADST bit in the A/D control register (ADCR) after the A/D start trigger select register (ADSTRGR) and the A/D analog input channel select register (ADANSR) have been set. Do not modify the settings of the ADCS, ACE, ADIE, TRGE, and EXTRG bits while the ADST bit in the ADCR register is set to 1.
- Do not start the A/D conversion when the ANS bits (ANS[7:0]) in the A/D analog input channel select register (ADANSR) are all 0.

17.7.7 Treatment of AVcc and AVss When the A/D Converter is Not Used

When the A/D converter is not used, it is recommended that AVcc be connected to VccQ and AVss be connected to VssQ.

Section 18 D/A Converter (DAC)

18.1 Features

- 8-bit resolution
- Two output channels
- Maximum conversion time of 10 µs (with 20 pF load)
- Output voltage of 0 V to AVREF
- D/A output hold function in software standby mode
- Module standby mode can be set

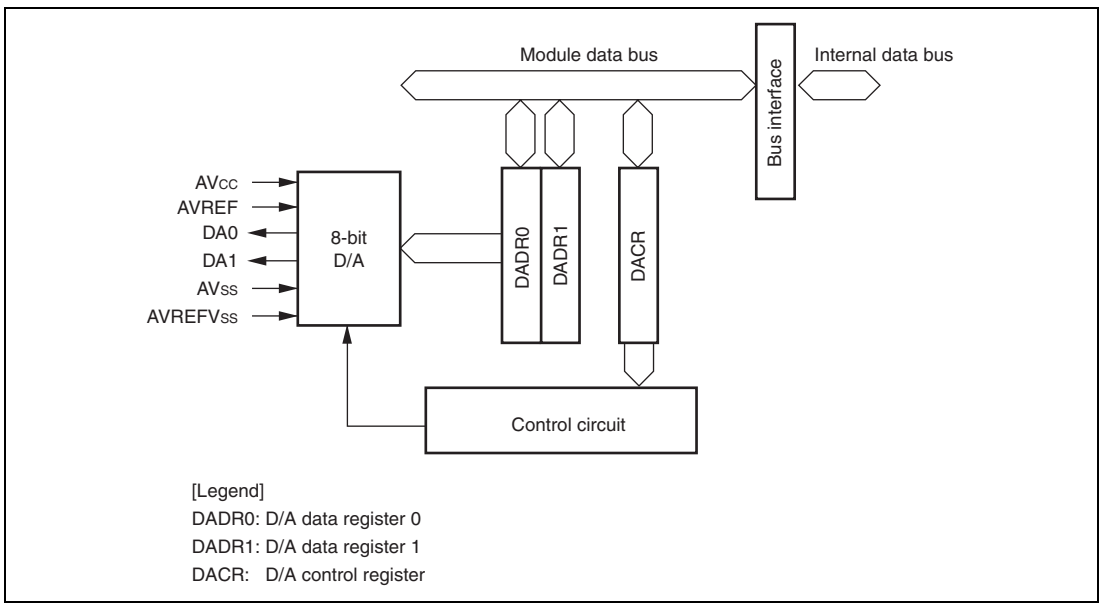


Figure 18.1 Block Diagram of D/A Converter

18.2 Input/Output Pins

Table 18.1 shows the pin configuration of the D/A converter.

Table 18.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog block power supply
Analog ground pin	AVss	Input	Analog block ground
Reference voltage pin	AVREF	Input	D/A conversion reference voltage
Reference ground pin	AVREFVss	Input	D/A conversion reference ground
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

18.3 Register Descriptions

The D/A converter has the following registers.

Table 18.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
D/A data register 0	DADR0	R/W	H'00	H'FFFE6800	8, 16
D/A data register 1	DADR1	R/W	H'00	H'FFFE6801	8, 16
D/A control register	DACR	R/W	H'1F	H'FFFE6802	8, 16

18.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1)

DADR is an 8-bit readable/writable register that stores data to which D/A conversion is to be performed. Whenever analog output is enabled, the values in DADR are converted and output to the analog output pins.

DADR is initialized to H'00 by a power-on reset or in module standby mode.

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

18.3.2 D/A Control Register (DACR)

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter.

DACR is initialized to H'1F by a power-on reset or in module standby mode.

Bit:	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	-	-	-	-	-
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R/W	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output for channel 1. 0: Analog output of channel 1 (DA1) is disabled 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output for channel 0. 0: Analog output of channel 0 (DA0) is disabled 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. Output of conversion results is always controlled by the DAOE0 and DAOE1 bits. For details, see table 18.3. 0: D/A conversion for channels 0 and 1 is controlled independently 1: D/A conversion for channels 0 and 1 is controlled together
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 18.3 Control of D/A Conversion

Bit 5	Bit 7	Bit 6	
DAE	**DAOE1**	**DAOE0**	**Description**
0	0	0	D/A conversion is disabled.
		1	D/A conversion of channel 0 is enabled and D/A conversion of channel 1 is disabled.
	1	0	D/A conversion of channel 1 is enabled and D/A conversion of channel 0 is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
1	0	0	D/A conversion is disabled.
		1	D/A conversion of channels 0 and 1 is enabled.
	1	0	
		1	

18.4 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOE bit in DACR is set to 1, D/A conversion is enabled and the conversion result is output.

An operation example of D/A conversion on channel 0 is shown below. Figure 18.2 shows the timing of this operation.

1. Write the conversion data to DADR0.
2. Set the DAOE0 bit in DACR to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time t_{DCONV} has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Contents of DADR}}{256} \times \text{AVref}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time t_{DCONV} has elapsed.
4. If the DAOE0 bit is cleared to 0, analog output is disabled.

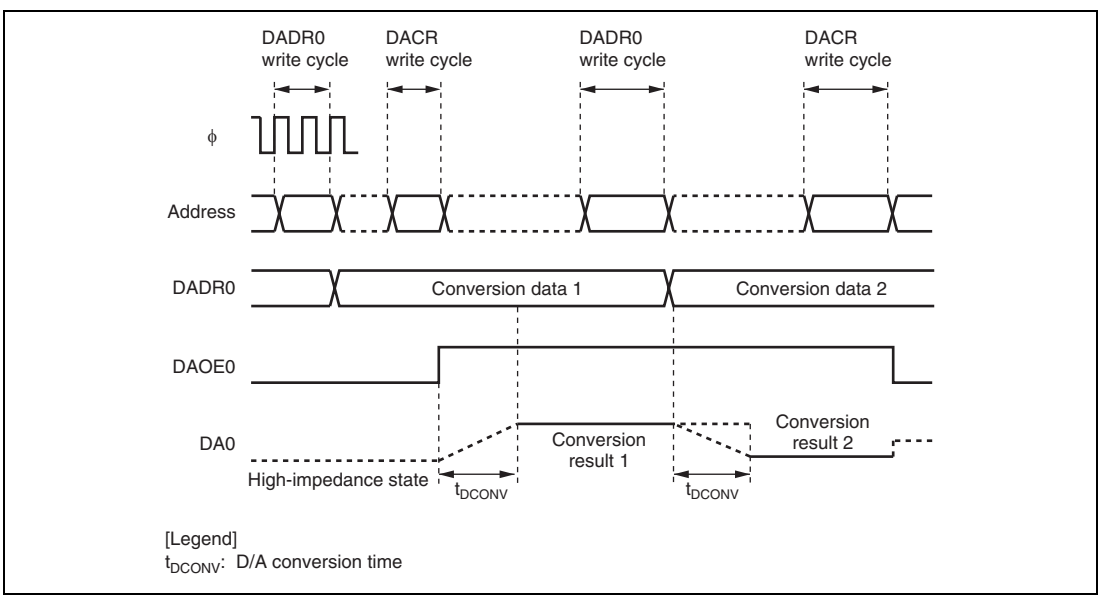


Figure 18.2 Example of D/A Converter Operation

18.5 Usage Notes

18.5.1 Module Standby Mode Setting

Operation of the D/A converter can be disabled or enabled using the standby control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by canceling module standby mode. For details, see section 23, Power-Down Modes.

18.5.2 D/A Output Hold Function in Software Standby Mode

When this LSI enters software standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is equal to as during D/A conversion. If the analog power supply current needs to be reduced in software standby mode, clear the DAOE0, DAOE1, and DAE bits to 0 to disable the D/A outputs.

18.5.3 Setting Analog Input Voltage

The reliability of this LSI may be adversely affected if the following voltage ranges are exceeded.

1. AVcc and AVss input voltages

   Input voltages AVcc and AVss should be Vcc ≤ AVcc ≤ 5.0 V ± 0.5 V and AVss = Vss. Do not leave the AVcc and AVss pins open when the A/D converter or D/A converter is not in use and in software standby mode. When not in use, connect AVcc to the power supply (VccQ) and AVss to the ground (VssQ).
2. Setting range of AVREF input voltage

   Set the voltage range of the AVREF pin as AVREF = AVcc ± 0.3 V when the A/D converter or D/A converter is used, or as AVREF = AVcc when no A/D converter or D/A converter is used.

Section 19 Pin Function Controller (PFC)

The pin function controller (PFC) is composed of registers that are used to select the functions of multiplexed pins and assign pins to be inputs or outputs. Tables 19.1 to 19.4 list the multiplexed pins of this LSI.

Table 19.1 Multiplexed Pins (Port A)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)
A	PA25 I/O (port)	A25 output (BSC)	—	IRQ7 input (INTC)	TIOC0D I/O (MTU2)	TXD1 output (SCIF1)	—
	PA24 I/O (port)	A24 output (BSC)	—	IRQ6 input (INTC)	TIOC0C I/O (MTU2)	RXD1 input (SCIF1)	—
	PA23 I/O (port)	A23 output (BSC)	—	IRQ5 input (INTC)	TIOC0B I/O (MTU2)	SCK1 I/O (SCIF1)	—
	PA22 I/O (port)	A22 output (BSC)	—	IRQ4 input (INTC)	TIOC0A I/O (MTU2)	—	—
	PA21 I/O (port)	A21 output (BSC)	—	IRQ3 input (INTC)	—	—	—
	PA20 I/O (port)	A20 output (BSC)	—	IRQ2 input (INTC)	—	—	—
	PA19 I/O (port)	A19 output (BSC)	—	IRQ1 input (INTC)	—	—	—
	PA18 I/O (port)	A18 output (BSC)	—	IRQ0 input (INTC)	—	—	—
	PA17 I/O (port)	A17 output (BSC)	—	—	—	TXD3 output (SCIF3)	—
	PA16 I/O (port)	A16 output (BSC)	—	—	—	RXD3 input (SCIF3)	—
	PA15 I/O (port)	A15 output (BSC)	—	—	—	SCK3 I/O (SCIF3)	—
	PA14 I/O (port)	A14 output (BSC)	—	—	—	—	—
	PA13 I/O (port)	A13 output (BSC)	—	—	—	—	—

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)
A	PA12 I/O (port)	A12 output (BSC)	—	—	—	—	—
	PA11 I/O (port)	A11 output (BSC)	—	—	—	—	—
	PA10 I/O (port)	A10 output (BSC)	—	—	—	—	—
	PA9 I/O (port)	A9 output (BSC)	—	—	—	—	—
	PA8 I/O (port)	A8 output (BSC)	—	—	—	—	—
	PA7 I/O (port)	A7 output (BSC)	—	—	—	—	—
	PA6 I/O (port)	A6 output (BSC)	—	—	—	—	—
	PA5 I/O (port)	A5 output (BSC)	—	—	—	—	—
	PA4 I/O (port)	A4 output (BSC)	—	—	—	—	—
	PA3 I/O (port)	A3 output (BSC)	—	—	—	—	—
	PA2 I/O (port)	A2 output (BSC)	—	—	—	—	—
	PA1 I/O (port)	A1 output (BSC)	—	—	—	—	—
	PA0 I/O (port)	A0 output (BSC)	—	—	—	—	—

Table 19.2 Multiplexed Pins (Port B)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)
B	PB30 I/O (port)	$\overline{\text{IRQOUT}}$/ $\overline{\text{REFOUT}}$ output (INTC/BSC)	—	$\overline{\text{UBCTRG}}$ output (UBC)	—	—	$\overline{\text{ASEBRKAK}}$/ $\overline{\text{ASEBRK}}$ output (AUD)
	PB29 I/O (port)	—	DREQ0 input (DMAC)	—	TIOC1B I/O (MTU2)	—	—
	PB28 I/O (port)	—	DACK0 output (DMAC)	—	TIOC1A I/O (MTU2)	RXD3 input (SCIF3)	—
	PB27 I/O (port)	—	TEND0 output (DMAC)	—	TIOC2A I/O (MTU2)	TXD3 output (SCIF3)	AUDATA0 output (AUD)
	PB26 I/O (port)	—	DREQ1 input (DMAC)	—	TIOC2B I/O (MTU2)	SCK3 I/O (SCIF3)	AUDATA1 output (AUD)
	PB25 I/O (port)	—	DACK1 output (DMAC)	IRQ3 input (INTC)	TCLKA input (MTU2)	TXD3 output (SCIF3)	AUDATA2 output (AUD)
	PB24 I/O (port)	—	TEND1 output (DMAC)	IRQ2 input (INTC)	TCLKB input (MTU2)	RXD3 input (SCIF3)	AUDATA3 output (AUD)
	PB23 I/O (port)	—	DREQ2 input (DMAC)	—	TCLKC input (MTU2)	TXD2 output (SCIF2)	AUDCK output (AUD)
	PB22 I/O (port)	—	DACK2 output (DMAC)	—	TCLKD input (MTU2)	RXD2 input (SCIF2)	AUDSYNC output (AUD)
	PB21 I/O (port)	$\overline{\text{CS2}}$ output (BSC)	—	IRQ0 input (INTC)	TIOC3BS I/O (MTU2S)	RXD0 input (SCIF0)	—
	PB20 I/O (port)	$\overline{\text{BS}}$ output (BSC)	—	—	TIOC3DS I/O (MTU2S)	—	—
	PB19 I/O (port)	$\overline{\text{CS6}}$ output (BSC)	—	IRQ6 input (INTC)	TIOC3D I/O (MTU2)	—	—
	PB18 I/O (port)	$\overline{\text{CS4}}$ output (BSC)	—	IRQ4 input (INTC)	TIOC3B I/O (MTU2)	—	—
	PB17 I/O (port)	$\overline{\text{CS3}}$ output (BSC)	—	IRQ1 input (INTC)	TIOC3A I/O (MTU2)	—	—

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)
B	PB16 I/O (port)	$\overline{CS1}$ output (BSC)	—	—	$\overline{POE1}$ input (POE)	TXD0 output (SCIF0)	—
	PB15 I/O (port)	$\overline{CS5}$ output (BSC)	—	IRQ5 input (INTC)	TIOC3C I/O (MTU2)	—	—
	PB14 I/O (port)	—	—	$\overline{ADTRG}$ input (ADC)	—	RXD2 input (SCIF2)	$\overline{MRES}$ input (system control)
	PB13 I/O (port)	$\overline{BACK}$ output (BSC)	—	—	TIOC4BS I/O (MTU2S)	SCK2 I/O (SCIF2)	—
	PB12 I/O (port)	$\overline{BREQ}$ input (BSC)	—	—	TIOC4AS I/O (MTU2S)	TXD2 output (SCIF2)	—
	PB11 I/O (port)	$\overline{AH}$ output (BSC)	DACK3 output (DMAC)	—	TIOC4DS I/O (MTU2S)	TXD2 output (SCIF2)	—
	PB10 I/O (port)	$\overline{WAIT}$ input (BSC)	DREQ3 input (DMAC)	—	TIOC4CS I/O (MTU2S)	RXD2 input (SCIF2)	—
	PB9 I/O (port)	$\overline{WE1}$/$\overline{DQMLU}$ output (BSC)	—	—	TIOC3CS I/O (MTU2S)	TXD3 (SCIF3)	—
	PB8 I/O (port)	$\overline{WE0}$/$\overline{DQMLL}$ output (BSC)	—	—	TIOC3AS I/O (MTU2S)	RXD3 (SCIF3)	—
	PB7 I/O (port)	$\overline{CS7}$ output (BSC)	—	IRQ7 input (INTC)	TIOC4D I/O (MTU2)	—	—
	PB6 I/O (port)	$\overline{CASL}$ output (BSC)	—	IRQ3 input (INTC)	TIOC4C I/O (MTU2)	—	—
	PB5 I/O (port)	$\overline{RASL}$ output (BSC)	—	IRQ2 input (INTC)	TIOC4B I/O (MTU2)	—	—
	PB4 I/O (port)	CKE output (BSC)	—	—	TIOC4A I/O (MTU2)	—	—
	PB3 I/O (port)	CK output (CPG)	—	—	—	—	—
	PB2 I/O (port)	$\overline{CS0}$ output (BSC)	—	—	$\overline{POE4}$ input (POE)	SCK0 I/O (SCIF0)	—
	PB1 I/O (port)	RD/$\overline{WR}$ output (BSC)	—	—	$\overline{POE8}$ input (POE)	TXD0 output (SCIF0)	—
	PB0 I/O (port)	$\overline{RD}$ output (BSC)	—	—	$\overline{POE0}$ input (POE)	RXD0 input (SCIF0)	—

Table 19.3 Multiplexed Pins (Port D)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)
D	PD15 I/O (port)	D15 I/O (BSC)	—	—	TIC5US input (MTU2S)	—	—
	PD14 I/O (port)	D14 I/O (BSC)	—	—	TIC5VS input (MTU2S)	—	—
	PD13 I/O (port)	D13 I/O (BSC)	—	—	TIC5WS input (MTU2S)	—	—
	PD12 I/O (port)	D12 I/O (BSC)	—	—	TIC5U input (MTU2)	—	—
	PD11 I/O (port)	D11 I/O (BSC)	—	—	TIC5V input (MTU2)	—	—
	PD10 I/O (port)	D10 I/O (BSC)	—	—	TIC5W input (MTU2)	—	—
	PD9 I/O (port)	D9 I/O (BSC)	—	—	—	—	—
	PD8 I/O (port)	D8 I/O (BSC)	—	—	—	—	—
	PD7 I/O (port)	D7 I/O (BSC)	—	—	—	—	—
	PD6 I/O (port)	D6 I/O (BSC)	—	—	—	—	—
	PD5 I/O (port)	D5 I/O (BSC)	—	—	—	—	—
	PD4 I/O (port)	D4 I/O (BSC)	—	—	—	—	—
	PD3 I/O (port)	D3 I/O (BSC)	—	—	—	—	—
	PD2 I/O (port)	D2 I/O (BSC)	—	—	—	—	—
	PD1 I/O (port)	D1 I/O (BSC)	—	—	—	—	—
	PD0 I/O (port)	D0 I/O (BSC)	—	—	—	—	—

Table 19.4 Multiplexed Pins (Port F)

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)	Function 5 (Related Module)	Function 6 (Related Module)	Function 7 (Related Module)
F	PF1 input (port)	—	—	IRQ1 input (INTC)	$\overline{\text{POE3}}$ input (POE)	SDA I/O (IIC3)	—
	PF0 input (port)	—	—	IRQ0 input (INTC)	$\overline{\text{POE7}}$ input (POE)	SCL I/O (IIC3)	—

Note: When function 7 of PB22 is selected, function 7 of PB23 to PB27 is automatically selected.

19.1 Register Descriptions

The PFC has the following registers.

Table 19.5 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 H3	PACRH3	R/W	H'0000/ H'0011	H'FFFE380A	8, 16
Port A control register H2	PACRH2	R/W	H'0000/ H'1111	H'FFFE380C	8, 16, 32
Port A control register H1	PACRH1	R/W	H'0000/ H'1111	H'FFFE380E	8, 16
Port A control register L4	PACRL4	R/W	H'0000/ H'1111	H'FFFE3810	8, 16, 32
Port A control register L3	PACRL3	R/W	H'0000/ H'1111	H'FFFE3812	8, 16
Port A control register L2	PACRL2	R/W	H'0000/ H'1111	H'FFFE3814	8, 16, 32
Port A control register L1	PACRL1	R/W	H'0000/ H'1111	H'FFFE3816	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 H4	PBCRH4	R/W	H'0000	H'FFFE3888	8, 16, 32
Port B control register H3	PBCRH3	R/W	H'0000	H'FFFE388A	8, 16
Port B control register H2	PBCRH2	R/W	H'0000	H'FFFE388C	8, 16, 32
Port B control register H1	PBCRH1	R/W	H'0000/ H'0001	H'FFFE388E	8, 16
Port B control register L4	PBCRL4	R/W	H'0000	H'FFFE3890	8, 16, 32
Port B control register L3	PBCRL3	R/W	H'0000/ H'0011	H'FFFE3892	8, 16
Port B control register L2	PBCRL2	R/W	H'0000	H'FFFE3894	8, 16, 32
Port B control register L1	PBCRL1	R/W	H'0000/ H'0101/ H'1101	H'FFFE3896	8, 16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D I/O register	PDIOR	R/W	H'0000	H'FFFE3986	8, 16
Port D control register L4	PDCRL4	R/W	H'0000/ H'1111	H'FFFE3990	8, 16, 32
Port D control register L3	PDCRL3	R/W	H'0000/ H'1111	H'FFFE3992	8, 16
Port D control register L2	PDCRL2	R/W	H'0000/ H'1111	H'FFFE3994	8, 16, 32
Port D control register L1	PDCRL1	R/W	H'0000/ H'1111	H'FFFE3996	8, 16
Port F control register L1	PFCRL1	R/W	H'0000	H'FFFE3A96	8, 16
IRQOUT function control register	IFCR	R/W	H'0000	H'FFFE38A2	16
WAVE function control register 2	WAVECR2	R/W	H'0001	H'FFFE3A14	8, 16, 32
WAVE function control register 1	WAVECR1	R/W	H'1111	H'FFFE3A16	8, 16

19.1.1 Port A I/O Registers H, L (PAIORH, PAIORL)

PAIORH and PAIORL are 16-bit readable/writable registers that are used to set the pins on port A as inputs or outputs. Bits PA25IOR to PA0IOR correspond to pins PA25 to PA0. PAIORH and PAIORL are enabled when the port A pins are functioning as general-purpose inputs/outputs (PA25 to PA0) and for the TIOC input/output of the MTU2. 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 10 of PAIORH are reserved. These bits are always read as 0. The write value should always be 0.

PAIORH and PAIORL are initialized to H'0000 by a power-on reset; but are not initialized by a manual reset or in sleep mode or software standby mode.

(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
	-	-	-	-	-	-	PA25 IOR	PA24 IOR	PA23 IOR	PA22 IOR	PA21 IOR	PA20 IOR	PA19 IOR	PA18 IOR	PA17 IOR	PA16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.1.2 Port A Control Registers H1 to H3, L1 to L4 (PACRH1 to PACRH3, PACRL1 to PACRL4)

PACRH1 to PACRH3 and PACRL1 to PACRL4 are 16-bit readable/writable registers that are used to select the functions of the multiplexed pins on port A.

PACRH1 to PACRH3 and PACRL1 to PACRL4 are initialized to the values shown in table 19.5 by a power-on reset; but are not initialized by a manual reset or in sleep mode or software standby mode.

(1) Port A Control Register H3 (PACRH3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PA25MD[2:0]			-	PA24MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 4	PA25MD[2:0]	000*	R/W	PA25 Mode Select the function of the PA25/A25/IRQ7/TIOC0D/TXD1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA25 I/O (port) 001: A25 output (BSC) (initial value) 010: Setting prohibited 011: IRQ7 input (INTC) 100: TIOC0D I/O (MTU2) 101: TXD1 output (SCIF) 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	PA24MD[2:0]	000*	R/W	PA24 Mode Select the function of the PA24/A24/IRQ6/TIOC0C/RXD1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA24 I/O (port) 001: A24 output (BSC) (initial value) 010: Setting prohibited 011: IRQ6 input (INTC) 100: TIOC0C I/O (MTU2) 101: RXD1 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(2) Port A Control Register H2 (PACRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA23MD[2:0]			-	PA22MD[2:0]			-	PA21MD[2:0]			-	PA20MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/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	PA23MD[2:0]	000*	R/W	PA23 Mode Select the function of the PA23/A23/IRQ5/TIOC0B/SCK1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA23 I/O (port) 001: A23 output (BSC) (initial value) 010: Setting prohibited 011: IRQ5 input (INTC) 100: TIOC0B I/O (MTU2) 101: SCK1 I/O (SCIF) 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	PA22MD[2:0]	000*	R/W	PA22 Mode Select the function of the PA22/A22/IRQ4/TIOC0A pin. • Area 0: 16-bit mode/8-bit mode 000: PA22 I/O (port) 001: A22 output (BSC) (initial value) 010: Setting prohibited 011: IRQ4 input (INTC) 100: TIOC0A 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	PA21MD[2:0]	000*	R/W	PA21 Mode Select the function of the PA21/A21/IRQ3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA21 I/O (port) 001: A21 output (BSC) (initial value) 010: Setting prohibited 011: IRQ3 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 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	PA20MD[2:0]	000*	R/W	PA20 Mode Select the function of the PA20/A20/IRQ2 pin. • Area 0: 16-bit mode/8-bit mode 000: PA20 I/O (port) 001: A20 output (BSC) (initial value) 010: Setting prohibited 011: IRQ2 input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(3) Port A Control Register H1 (PACRH1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA19MD[2:0]			-	PA18MD[2:0]			-	PA17MD[2:0]			-	PA16MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	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	PA19MD[2:0]	000*	R/W	PA19 Mode Select the function of the PA19/A19/IRQ1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA19 I/O (port) 001: A19 output (BSC) (initial value) 010: Setting prohibited 011: IRQ1 input (INTC) 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	PA18MD[2:0]	000*	R/W	PA18 Mode Select the function of the PA18/A18/IRQ0 pin. • Area 0: 16-bit mode/8-bit mode 000: PA18 I/O (port) 001: A18 output (BSC) (initial value) 010: Setting prohibited 011: IRQ0 input (INTC) 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	PA17MD[2:0]	000*	R/W	PA17 Mode Select the function of the PA17/A17/TXD3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA17 I/O (port) 001: A17 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: TXD3 output (SCIF) 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 Select the function of the PA16/A16/RXD3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA16 I/O (port) 001: A16 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: RXD3 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(4) Port A Control Register L4 (PACRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15MD[2:0]			-	PA14MD[2:0]			-	PA13MD[2:0]			-	PA12MD[2:0]		
Initial value:	0	0	0	0	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	PA15MD[2:0]	000*	R/W	PA15 Mode Select the function of the PA15/A15/SCK3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA15 I/O (port) 001: A15 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: SCK3 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PA14MD[2:0]	000*	R/W	PA14 Mode Select the function of the PA14/A14 pin. • Area 0: 16-bit mode/8-bit mode 000: PA14 I/O (port) 001: A14 output (BSC) (initial value) 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	PA13MD[2:0]	000*	R/W	PA13 Mode Select the function of the PA13/A13 pin. • Area 0: 16-bit mode/8-bit mode 000: PA13 I/O (port) 001: A13 output (BSC) (initial value) 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	PA12MD[2:0]	000*	R/W	PA12 Mode Select the function of the PA12/A12 pin. • Area 0: 16-bit mode/8-bit mode 000: PA12 I/O (port) 001: A12 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(5) Port A Control Register L3 (PACRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11MD[2:0]			-	PA10MD[2:0]			-	PA9MD[2:0]			-	PA8MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA11MD[2:0]	000*	R/W	PA11 Mode Select the function of the PA11/A11 pin. • Area 0: 16-bit mode/8-bit mode 000: PA11 I/O (port) 001: A11 output (BSC) (initial value) 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	PA10MD[2:0]	000*	R/W	PA10 Mode Select the function of the PA10/A10 pin. • Area 0: 16-bit mode/8-bit mode 000: PA10 I/O (port) 001: A10 output (BSC) (initial value) 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	PA9MD[2:0]	000*	R/W	PA9 Mode Select the function of the PA9/A9 pin. • Area 0: 16-bit mode/8-bit mode 000: PA9 I/O (port) 001: A9 output (BSC) (initial value) 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	PA8MD[2:0]	000*	R/W	PA8 Mode Select the function of the PA8/A8 pin. • Area 0: 16-bit mode/8-bit mode 000: PA8 I/O (port) 001: A8 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(6) Port A Control Register L2 (PACRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7MD[2:0]			-	PA6MD[2:0]			-	PA5MD[2:0]			-	PA4MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA7MD[2:0]	000*	R/W	PA7 Mode Select the function of the PA7/A7 pin. • Area 0: 16-bit mode/8-bit mode 000: PA7 I/O (port) 001: A7 output (BSC) (initial value) 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	PA6MD[2:0]	000*	R/W	PA6 Mode Select the function of the PA6/A6 pin. • Area 0: 16-bit mode/8-bit mode 000: PA6 I/O (port) 001: A6 output (BSC) (initial value) 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	PA5MD[2:0]	000*	R/W	PA5 Mode Select the function of the PA5/A5 pin. • Area 0: 16-bit mode/8-bit mode 000: PA5 I/O (port) 001: A5 output (BSC) (initial value) 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	PA4MD[2:0]	000*	R/W	PA4 Mode Select the function of the PA4/A4 pin. • Area 0: 16-bit mode/8-bit mode 000: PA4 I/O (port) 001: A4 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(7) Port A Control Register L1 (PACRL1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3MD[2:0]			-	PA2MD[2:0]			-	PA1MD[2:0]			-	PA0MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PA3MD[2:0]	000*	R/W	PA3 Mode Select the function of the PA3/A3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA3 I/O (port) 001: A3 output (BSC) (initial value) 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	PA2MD[2:0]	000*	R/W	PA2 Mode Select the function of the PA2/A2 pin. • Area 0: 16-bit mode/8-bit mode 000: PA2 I/O (port) 001: A2 output (BSC) (initial value) 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	PA1MD[2:0]	000*	R/W	PA1 Mode Select the function of the PA1/A1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA1 I/O (port) 001: A1 output (BSC) (initial value) 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	PA0MD[2:0]	000*	R/W	PA0 Mode Select the function of the PA0/A0 pin. • Area 0: 16-bit mode/8-bit mode 000: PA0 I/O (port) 001: A0 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

19.1.3 Port B I/O Registers 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. Bits PB30IOR to PB0IOR correspond to pins PB30 to PB0, respectively. PBIOR is enabled when the port B pins are functioning as general-purpose inputs/outputs (PB9, PB5, and PB4). In other states, PBIOR is disabled. A given pin on port B will be an output pin if the corresponding bit in PBIORH and PBIORL is set to 1, and an input pin if the bit is cleared to 0.

Bit 15 of PBIORH is reserved. This bit is always read as 0. The write value should always be 0.

PBIORH and PBIORL are initialized to H'0000 by a power-on reset; but are not initialized by a manual reset or in sleep mode or software standby mode.

(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
	-	PB30 IOR	PB29 IOR	PB28 IOR	PB27 IOR	PB26 IOR	PB25 IOR	PB24 IOR	PB23 IOR	PB22 IOR	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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	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
	PB15 IOR	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

19.1.4 Port B Control Registers H1 to H4, L1 to L4 (PBCRH1 to PBCRH4, PBCRL1 to PBCRL4)

PBCRH1 to PBCRH4 and PBCRL1 to PBCRL4 are 16-bit readable/writable registers that are used to select the function of the multiplexed pins on port B.

PBCRH1 to PBCRH4 and PBCRL1 to PBCRL4 are initialized to the values shown in table 19.5 by a power-on reset; but are not initialized by a manual reset or in sleep mode or software standby mode.

(1) Port B Control Register H4 (PBCRH4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PB30MD[2:0]			-	PB29MD[2:0]			-	PB28MD[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

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	PB30MD[2:0]	000	R/W	PB30 Mode Select the function of the PB30/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/$\overline{\text{UBCTRG}}$ pin. 000: PB30 I/O (port) 001: $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output (INTC/BSC) 010: Setting prohibited 011: $\overline{\text{UBCTRG}}$ output (UBC) 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	PB29MD[2:0]	000	R/W	PB29 Mode Select the function of the PB29/DREQ0/TIOC1B pin. 000: PB29 I/O (port) 001: Setting prohibited 010: DREQ0 input (DMAC) 011: Setting prohibited 100: TIOC1B I/O (MTU2) 101: RXD3 input (SCIF) 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	PB28MD[2:0]	000	R/W	PB28 Mode Select the function of the PB28/DACK0/TIOC1A/RXD3 pin. 000: PB28 I/O (port) 001: Setting prohibited 010: Dack0 output (DMAC) 011: Setting prohibited 100: Setting prohibited 101: TIOC1A I/O (MTU2) 110: Setting prohibited 111: Setting prohibited

(2) Port B Control Register H3 (PBCRH3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB27MD[2:0]			-	PB26MD[2:0]			-	PB25MD[2:0]			-	PB24MD[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	PB27MD[2:0]	000	R/W	PB27 Mode Select the function of the PB27/TEND0/TIOC2A/TXD3/AUDATA0 pin. 000: PB27 I/O (port) 001: Setting prohibited 010: TEND0 output (DMAC) 011: Setting prohibited 100: TIOC2A I/O (MTU2) 101: TXD3 output (SCIF) 110: AUDATA0 output (AUD) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB26MD[2:0]	000	R/W	PB26 Mode Select the function of the PB26/DREQ0/TIOC2B/SCK3/AUDATA1 pin. 000: PB26 I/O (port) 001: Setting prohibited 010: DREQ1 input (DMAC) 011: Setting prohibited 100: TIOC2B I/O (MTU2) 101: SCK3 I/O (SCIF) 110: AUDATA1 output (AUD) 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	PB25MD[2:0]	000	R/W	PB25 Mode Select the function of the PB25/DACK1/IRQ3/TCLKA/TXD3/AUDATA2 pin. 000: PB25 I/O (port) 001: Setting prohibited 010: DACK1 output (DMAC) 011: IRQ3 input (INTC) 100: TCLKA input (MTU2) 101: TXD3 output (SCIF) 110: AUDATA2 output (AUD) 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PB24MD[2:0]	000	R/W	PB24 Mode Select the function of the PB24/TEND1/IRQ2/TCLKB/RXD3/AUDATA3 pin. 000: PB24 I/O (port) 001: Setting prohibited 010: TEND1 output (DMAC) 011: IRQ2 input (INTC) 100: TCLKB input (MTU2) 101: RXD3 input (SCIF) 110: AUDATA3 output (AUD) 111: Setting prohibited.

(3) Port B Control Register H2 (PBCRH2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB23MD[2:0]			-	PB22MD[2:0]			-	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/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	PB23MD[2:0]	000	R/W	PB23 Mode Select the function of the PB23/DREQ2/TCLKC/TXD2/AUDCK pin. 000: PB23 I/O (port) 001: Setting prohibited 010: DREQ2 input (DMAC) 011: Setting prohibited 100: TCLKC input (MTU2) 101: TXD2 output (SCIF) 110: AUDCK output (AUD) 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB22MD[2:0]	000	R/W	PB22 Mode Select the function of the PB22/DACK2/TCLKD/RXD2/AUDSYNC pin. 000: PB22 I/O (port) 001: Setting prohibited 010: DACK2 output (DMAC) 011: Setting prohibited 100: TCLKD input (MTU2) 101: RXD2 input (SCIF) 110: AUDSYNC output (AUD) 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	PB21MD[2:0]	000	R/W	PB21 Mode Select the function of the PB21/$\overline{CS2}$/IRQ0/TIOC3BS/RXD0 pin. 000: PB21 I/O (port) 001: $\overline{CS2}$ output (BSC) 010: Setting prohibited 011: IRQ0 input (INTC) 100: TIOC3BS I/O (MTU2S) 101: RXD0 input (SCIF) 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	PB20MD[2:0]	000	R/W	PB20 Mode Select the function of the PB20/$\overline{BS}$/TIOC3DS pin. 000: PB20 I/O (port) 001: $\overline{BS}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC3DS I/O (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

(4) 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 Select the function of the PB19/$\overline{\text{CS6}}$/IRQ6/TIOC3D pin. 000: PB19 I/O (port) 001: $\overline{\text{CS6}}$ output (BSC) 010: Setting prohibited 011: IRQ6 input (INTC) 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	PB18MD[2:0]	000	R/W	PB18 Mode Select the function of the PB18/$\overline{\text{CS4}}$/IRQ4/TIOC3B pin. 000: PB18 I/O (port) 001: $\overline{\text{CS4}}$ output (BSC) 010: Setting prohibited 011: IRQ4 input (INTC) 100: TIOC3B I/O (MTU2) 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	PB17MD[2:0]	000	R/W	PB17 Mode Select the function of the PB17/$\overline{CS3}$/IRQ1/TIOC3A pin. 000: PB17 I/O (port) 001: $\overline{CS3}$ output (BSC) 010: Setting prohibited 011: IRQ1 input (INTC) 100: TIOC3A 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	PB16MD[2:0]	000*	R/W	PB16 Mode Select the function of the PB16/$\overline{CS1}$/$\overline{POE1}$/TXD0 pin. • Area 0: 16-bit mode/8-bit mode 000: PB16 I/O (port) 001: $\overline{CS1}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE1}$ input (POE2) 101: TXD0 output (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(5) Port B Control Register L4 (PBCRL4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB15MD[2:0]			-	PB14MD[2:0]			-	PB13MD[2:0]			-	PB12MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PB15MD[2:0]	000	R/W	PB15 Mode Select the function of the PB15/$\overline{\text{CS5}}$/IRQ5/TIOC3C pin. 000: PB15 I/O (port) 001: $\overline{\text{CS5}}$ output (BSC) 010: Setting prohibited 011: IRQ5 input (INTC) 100: TIOC3C 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	PB14MD[2:0]	000	R/W	PB14 Mode Select the function of the PB14/$\overline{\text{ADTRG}}$/RXD2/$\overline{\text{MRES}}$ pin. 000: PB14 I/O (port) 001: Setting prohibited 010: Setting prohibited 011: $\overline{\text{ADTRG}}$ input (ADC) 100: Setting prohibited 101: RXD2 input (SCIF) 110: $\overline{\text{MRES}}$ input (system control) 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	PB13MD[2:0]	000	R/W	PB13 Mode Select the function of the PB13/$\overline{\text{BACK}}$/TIOC4BS/SCK2 pin. 000: PB13 I/O (port) 001: $\overline{\text{BACK}}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC4BS I/O (MTU2S) 101: SCK2 I/O (SCIF) 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	PB12MD[2:0]	000	R/W	PB12 Mode Select the function of the PB12/$\overline{\text{BREQ}}$/TIOC4AS/TXD2 pin. 000: PB12 I/O (port) 001: $\overline{\text{BREQ}}$ input (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC4AS I/O (MTU2S) 101: TXD2 output (SCIF) 110: Setting prohibited 111: Setting prohibited

(6) Port B Control Register L3 (PBCRL3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB11MD[2:0]			-	PB10MD[2:0]			-	PB9MD[2:0]			-	PB8MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PB11MD[2:0]	000	R/W	PB11 Mode Select the function of the PB11/$\overline{\text{AH}}$/DACK3/TIOC4DS/TXD2 pin. 000: PB11 I/O (port) 001: $\overline{\text{AH}}$ output (BSC) 010: DACK3 output (DMAC) 011: Setting prohibited 100: TIOC4DS I/O (MTU2S) 101: TXD2 output (SCIF) 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	PB10MD[2:0]	000	R/W	PB10 Mode Select the function of the PB10/$\overline{\text{WAIT}}$/DREQ3/TIOC4CS/RXD2 pin. 000: PB10 I/O (port) 001: $\overline{\text{WAIT}}$ input (BSC) 010: DREQ3 input (DMAC) 011: Setting prohibited 100: TIOC4CS I/O (MTU2S) 101: RXD2 input (SCIF) 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	PB9MD[2:0]	000*	R/W	PB9 Mode Select the function of the PB9/$\overline{WE1}$/$\overline{DQMLU}$/TIOC3CS/TXD3 pin. • Area 0: 16-bit mode/8-bit mode 000: PB9 I/O (port) 001: $\overline{WE1}$/$\overline{DQMLU}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIOC3CS I/O (MTU2S) 101: TXD3 output (SCIF) 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	PB8MD[2:0]	000*	R/W	PB8 Mode Select the function of the PB8/$\overline{WE0}$/$\overline{DQMLL}$/TIOC3AS/RXD3 pin. • Area 0: 16-bit mode/8-bit mode 000: PB8 I/O (port) 001: $\overline{WE0}$/$\overline{DQMLL}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIOC3AS I/O (MTU2S) 101: RXD3 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(7) Port B Control Register L2 (PBCRL2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB7MD[2:0]			-	PB6MD[2:0]			-	PB5MD[2:0]			-	PB4MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	PB7MD[2:0]	000	R/W	PB7 Mode Select the function of the PB7/$\overline{CS7}$/IRQ7/TIOC4D pin. 000: PB7 I/O (port) 001: $\overline{CS7}$ output (BSC) 010: Setting prohibited 011: IRQ7 input (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.
10 to 8	PB6MD[2:0]	000	R/W	PB6 Mode Select the function of the PB6/$\overline{CASL}$/IRQ3/TIOC4C pin. 000: PB6 I/O (port) 001: $\overline{CASL}$ output (BSC) 010: Setting prohibited 011: IRQ3 input (INTC) 100: TIOC4C I/O (MTU2) 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	PB5MD[2:0]	000	R/W	PB5 Mode Select the function of the PB5/$\overline{\text{RASL}}$/IRQ2/TIOC4B pin. 000: PB5 I/O (port) 001: $\overline{\text{RASL}}$ output (BSC) (initial value) 010: Setting prohibited 011: IRQ2 input (INTC) 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	PB4MD[2:0]	000	R/W	PB4 Mode Select the function of the PB4/CKE/TIOC4A pin. 000: PB4 I/O (port) 001: CKE output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC4A I/O (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

(8) 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 Select the function of the PB3/CK pin. • Area 0: 16-bit mode/8-bit mode 000: PB3 I/O (port) 001: CK output (CPG) (initial value) 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	PB2MD[2:0]	000*	R/W	PB2 Mode Select the function of the PB2/$\overline{CS0}$/$\overline{POE4}$/SCK0 pin. • Area 0: 16-bit mode/8-bit mode 000: PB2 I/O (port) 001: $\overline{CS0}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE4}$ input (POE2) 101: SCK0 I/O (SCIF) 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	PB1MD[2:0]	000	R/W	PB1 Mode Select the function of the PB1/RD/$\overline{WR}$/$\overline{POE8}$/TXD0 pin. 000: PB1 I/O (port) 001: RD/$\overline{WR}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE8}$ input (POE2) 101: TXD0 output (SCIF) 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	PB0MD[2:0]	000*	R/W	PB0 Mode Select the function of the PB0/$\overline{RD}$/$\overline{POE0}$/RXD0 pin. • Area 0: 16-bit mode/8-bit mode 000: PB0 I/O (port) 001: $\overline{RD}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE0}$ input (POE2) 101: RXD0 input (SCIF) 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

19.1.5 Port D I/O Register (PDIOR)

PDIOR is a 16-bit readable/writable register that is used to set the pins on port D as inputs or outputs. Bits PD15IOR to PD0IOR correspond to pins PD15 to PD0. PDIOR is enabled when the port D pins are functioning as general-purpose inputs/outputs (PD15 to PD0). In other states, PDIOR is disabled. A given pin on port D will be an output pin if the corresponding bit in PDIOR is set to 1, and an input pin if the bit is cleared to 0.

PDIOR is initialized to H'0000 by a power-on; but is not initialized by a manual reset or in sleep mode or software standby mode.

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

19.1.6 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.

PDCRL1 to PDCRL4 are initialized to the values shown in table 19.6 by a power-on reset; but are not initialized by a manual reset or in sleep mode or software standby mode.

Table 19.6 Initial Values of Port D Control Registers

	Initial Value	
Register Name	**Area 0: 16-Bit Mode**	**Area 0: 8-Bit Mode**
PDCRL4	H'1111	H'0000
PDCRL3	H'1111	H'0000
PDCRL2	H'1111	H'1111
PDCRL1	H'1111	H'1111

(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 Select the function of the PD15/D15/TIC5US pin. • Area 0: 16-bit mode 000: PD15 I/O (port) 001: D15 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5US input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD15 I/O (port) (initial value) 001: D15 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5US input (MTU2S) 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	PD14MD[2:0]	000*	R/W	PD14 Mode Select the function of the PD14/D14/TIC5VS pin. • Area 0: 16-bit mode 000: PD14 I/O (port) 001: D14 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5VS input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD14 I/O (port) (initial value) 001: D14 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5VS input (MTU2S) 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	PD13MD[2:0]	000*	R/W	PD13 Mode Select the function of the PD13/D13/TIC5WS pin. • Area 0: 16-bit mode 000: PD13 I/O (port) 001: D13 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5WS input (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD13 I/O (port) (initial value) 001: D13 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5WS input (MTU2S) 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	PD12MD[2:0]	000*	R/W	PD12 Mode Select the function of the PD12/D12/TIC5U pin. • Area 0: 16-bit mode 000: PD12 I/O (port) 001: D12 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5U input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD12 I/O (port) (initial value) 001: D12 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5U input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(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 Select the function of the PD11/D11/TIC5V pin. • Area 0: 16-bit mode 000: PD11 I/O (port) 001: D11 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5V input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD11 I/O (port) (initial value) 001: D11 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5V input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PD10MD[2:0]	000*	R/W	PD10 Mode Select the function of the PD10/D10/TIC5W pin. • Area 0: 16-bit mode 000: PD10 I/O (port) 001: D10 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: TIC5W input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD10 I/O (port) (initial value) 001: D10 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: TIC5W input (MTU2) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PD9MD[2:0]	000*	R/W	PD9 Mode Select the function of the PD9/D9 pin. • Area 0: 16-bit mode 000: PD9 I/O (port) 001: D9 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD9 I/O (port) (initial value) 001: D9 I/O (data) 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.

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PD8MD[2:0]	000*	R/W	PD8 Mode Select the function of the PD8/D8 pin. • Area 0: 16-bit mode 000: PD8 I/O (port) 001: D8 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 8-bit mode 000: PD8 I/O (port) (initial value) 001: D8 I/O (data) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(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 Select the function of the PD7/D7 pin. • Area 0: 8-bit mode 000: PD7 I/O (port) 001: D7 I/O (data) (initial value) 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	PD6MD[2:0]	000*	R/W	PD6 Mode Select the function of the PD6/D6 pin. • Area 0: 8-bit mode 000: PD6 I/O (port) 001: D6 I/O (data) (initial value) 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	PD5MD[2:0]	000*	R/W	PD5 Mode Select the function of the PD5/D5 pin. • Area 0: 8-bit mode 000: PD5 I/O (port) 001: D5 I/O (data) (initial value) 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	PD4MD[2:0]	000*	R/W	PD4 Mode Select the function of the PD4/D4 pin. • Area 0: 8-bit mode 000: PD4 I/O (port) 001: D4 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

(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 Select the function of the PD3/D3 pin. • Area 0: 8-bit mode 000: PD3 I/O (port) 001: D3 I/O (data) (initial value) 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	PD2MD[2:0]	000*	R/W	PD2 Mode Select the function of the PD2/D2 pin. • Area 0: 8-bit mode 000: PD2 I/O (port) 001: D2 I/O (data) (initial value) 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	PD1MD[2:0]	000*	R/W	PD1 Mode Select the function of the PD1/D1 pin. • Area 0: 8-bit mode 000: PD1 I/O (port) 001: D1 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	PD0MD[2:0]	000*	R/W	PD0 Mode Select the function of the PD0/D0 pin. • Area 0: 8-bit mode 000: PD0 I/O (port) 001: D0 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the operating mode of the LSI.

19.1.7 Port F Control Register L1 (PFCRL1)

PFCRL1 is a 16-bit readable/writable register that is used to select the function of the multiplexed pins on port F.

PFCRL1 is initialized to the value shown in table 19.5 by a power-on reset; but is not initialized by a manual reset or in sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PF1MD[2:0]			-	PF0MD[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
6 to 4	PF1MD[2:0]	000	R/W	PF1 Mode Select the function of the PF1/IRQ1/$\overline{\text{POE3}}$/SDA pin. 000: PF1 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ1 input (INTC) 100: $\overline{\text{POE3}}$ input (POE2) 101: SDA I/O (IIC3) 110: Setting prohibited 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	PF0MD[2:0]	000	R/W	PF0 Mode Select the function of the PF0/IRQ0/$\overline{\text{POE7}}$/SCL pin. 000: PF0 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ0 input (INTC) 100: $\overline{\text{POE7}}$ input (POE2) 101: SCL I/O (IIC3) 110: Setting prohibited 111: Setting prohibited

19.1.8 IRQOUT Function Control Register (IFCR)

IFCR is a 16-bit readable/writable register that is used to control the $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ pin output when it is selected as the multiplexed pin function by port B control register H4 (PBCRH4). When PBCRH4 selects another function, the IFCR setting does not affect the pin function.

IFCR is initialized to H'0000 by a power-on reset; but is not initialized by a manual reset or in sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	IRQMD[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	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, 0	IRQMD[1:0]	00	R/W	IRQOUT Mode Select the function of the $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ pin when bits 10 to 8 (PB30MD[2:0]) in PBCRH4 are set to B'001. 00: Interrupt request accept signal output 01: Refresh signal output 10: Interrupt request accept signal output or refresh signal output (depends on the operating state) 11: Always high-level output

19.1.9 WAVE Function Control Registers 1, 2 (WAVECR1, WAVECR2)

WAVECR1 and WAVECR2 are 16-bit readable/writable registers that are used to enable the WAVE pin functions.

WAVECR1 and WAVECR2 are initialized to H'1111 and H'0001 respectively by a power-on reset; but are not initialized by a manual reset or in sleep mode or software standby mode.

- WAVE Function Control Register 2 (WAVECR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	WVRMD[2:0]		
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	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	WVRMD[2:0]	001	R/W	WRXD Mode Select the function of the WRXD pin. 000: Setting prohibited 001: Initial value 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: WRXD input 111: Setting prohibited

- WAVE Function Control Register 1 (WAVECR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	WVTMD[2:0]			-	WVSMD[2:0]			-	-	-	-	-	-	-	-
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
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	WVTMD[2:0]	001	R/W	WTXD Mode Select the function of the WTXD pin. 000: Setting prohibited 001: Initial value 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: WTXD output 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	WVSMD[2:0]	001	R/W	WSCK Mode Select the function of the WSCK pin. 000: Setting prohibited 001: Initial value 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: WSCK output 111: Setting prohibited

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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

Section 20 I/O Ports

This LSI has four ports: A, B, D, and F.

All port pins are multiplexed with other pin functions. The functions of the multiplex pins are selected by means of the pin function controller (PFC).

Each port is provided with data registers for storing the pin data and port registers for reading the states of the pins.

20.1 Port A

Port A is an input/output port with the 26 pins shown in figure 20.1.

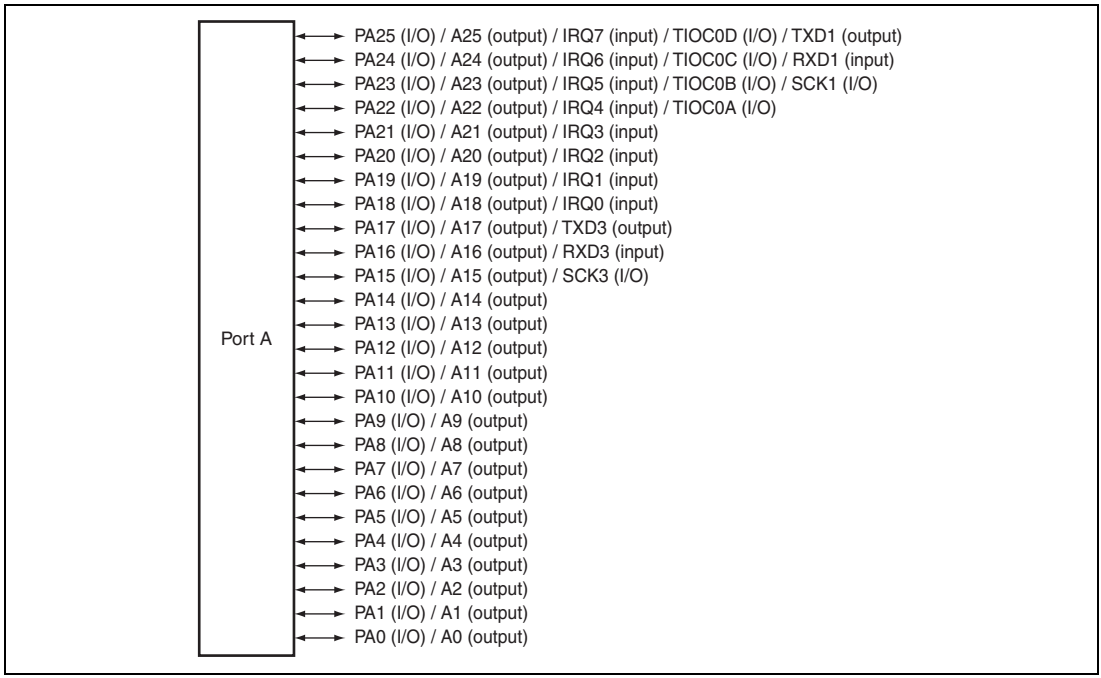


Figure 20.1 Port A

20.1.1 Register Descriptions

Table 20.1 lists the port A registers.

Table 20.1 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'xxxx	H'FFFE381C	8, 16, 32
Port A port register L	PAPRL	R	H'xxxx	H'FFFE381E	8, 16

20.1.2 Port A Data Registers H, L (PADRH, PADRL)

PADRH and PADRL are 16-bit readable/writable registers that store port A data. Bits PA25DR to PA0DR correspond to pins PA25 to PA0, respectively.

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 20.2 summarizes PADRH and PADRL read/write operations.

PADRH and PADRL are initialized to the respective values shown in table 20.1 by a power-on reset. PADRH and PADRL are not initialized by a manual reset or in sleep mode or software standby mode.

(1) Port A Data Register H (PADRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PA25DR	PA24DR	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
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/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 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	PA25DR	0	R/W	See table 20.2.
8	PA24DR	0	R/W	
7	PA23DR	0	R/W	
6	PA22DR	0	R/W	
5	PA21DR	0	R/W	
4	PA20DR	0	R/W	
3	PA19DR	0	R/W	
2	PA18DR	0	R/W	
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	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PA15DR	0	R/W	See table 20.2.
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

Table 20.2 Port A Data Registers H and L (PADRH and PADRL) Read/Write Operations

- PADRH bits 9 to 0 and PADRL bits 15 to 0

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.

20.1.3 Port A Port Registers H, L (PAPRH, PAPRL)

PAPRH and PAPRL are 16-bit read-only registers, in which bits PA25PR to PA0PR correspond to pins PA25 to PA0, respectively. PAPRH and PAPRL always return the states of the pins regardless of the PFC setting.

(1) Port A Port Register H (PAPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PA25PR	PA24PR	PA23PR	PA22PR	PA21PR	PA20PR	PA19PR	PA18PR	PA17PR	PA16PR
Initial value:	0	0	0	0	0	0	PA25	PA24	PA23	PA22	PA21	PA20	PA19	PA18	PA17	PA16
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 10	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
9	PA25PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
8	PA24PR	Pin state	R	
7	PA23PR	Pin state	R	
6	PA22PR	Pin state	R	
5	PA21PR	Pin state	R	
4	PA20PR	Pin state	R	
3	PA19PR	Pin state	R	
2	PA18PR	Pin state	R	
1	PA17PR	Pin state	R	
0	PA16PR	Pin state	R	

(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	PA14 PR	PA13 PR	PA12 PR	PA11 PR	PA10 PR	PA9 PR	PA8 PR	PA7 PR	PA6 PR	PA5 PR	PA4 PR	PA3 PR	PA2 PR	PA1 PR	PA0 PR
Initial value:	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
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	PA14PR	Pin state	R	
13	PA13PR	Pin state	R	
12	PA12PR	Pin state	R	
11	PA11PR	Pin state	R	
10	PA10PR	Pin state	R	
9	PA9PR	Pin state	R	
8	PA8PR	Pin state	R	
7	PA7PR	Pin state	R	
6	PA6PR	Pin state	R	
5	PA5PR	Pin state	R	
4	PA4PR	Pin state	R	
3	PA3PR	Pin state	R	
2	PA2PR	Pin state	R	
1	PA1PR	Pin state	R	
0	PA0PR	Pin state	R	

20.2 Port B

Port B is an input/output port with the 31 pins shown in figure 20.2.

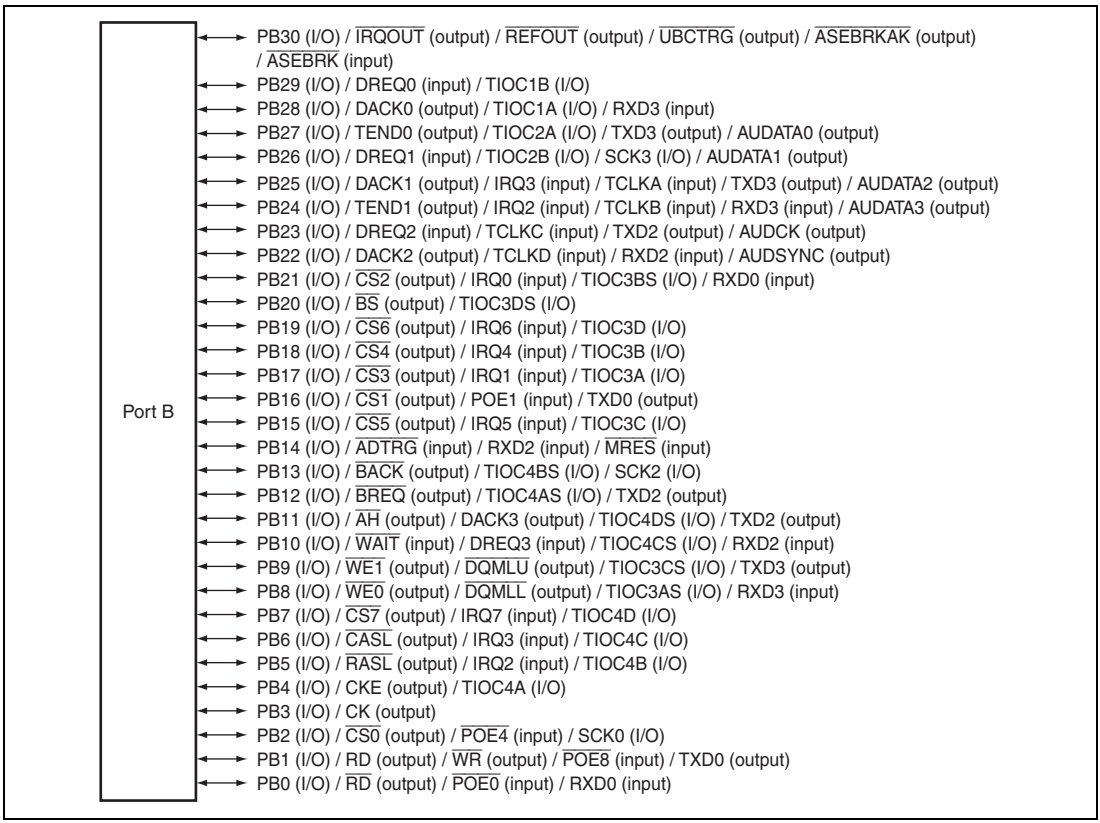


Figure 20.2 Port B

20.2.1 Register Descriptions

Table 20.3 lists the port B registers.

Table 20.3 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	PBPRH	R	H'xxxx	H'FFFE389C	8, 16, 32
Port B port register L	PBPRL	R	H'xxxx	H'FFFE389E	8, 16

20.2.2 Port B Data Registers H, L (PBDRH, PBDRL)

PBDRH and PBDRL are 16-bit readable/writable registers that store port B data. Bits PB30DR and PB0DR correspond to pins PB30 to PB0, respectively.

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 20.4 summarizes PBDRH and PBDRL read/write operations.

PBDRH and PBDRL are initialized to the value shown in table 20.3 by a power-on reset, but are not initialized by a manual reset or in sleep mode or software standby mode.

(1) Port B Data Register H (PBDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB30DR	PB29DR	PB28DR	PB27DR	PB26DR	PB25DR	PB24DR	PB23DR	PB22DR	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/W	R/W	R/W	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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PB30DR	0	R/W	See table 20.4.
13	PB29DR	0	R/W	
12	PB28DR	0	R/W	
11	PB27DR	0	R/W	
10	PB26DR	0	R/W	
9	PB25DR	0	R/W	
8	PB24DR	0	R/W	
7	PB23DR	0	R/W	
6	PB22DR	0	R/W	
5	PB21DR	0	R/W	
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
	PB15 DR	PB14 DR	PB13 DR	PB12 DR	PB11 DR	PB10 DR	PB9 DR	PB8 DR	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	PB0 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	PB15DR	0	R/W	See table 20.4.
14	PB14DR	0	R/W	
13	PB13DR	0	R/W	
12	PB12DR	0	R/W	
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

Table 20.4 Port B Data Registers H and L (PBDRH and PBDRL) Read/Write Operations

- PBDRH bits 14 to 0 and PBDRL bits 15 to 0

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

20.2.3 Port B Port Registers H, L (PBPRH, PBPRL)

PBPRH and PBPRL are 16-bit read-only registers, in which bits PB30PR to PB0PR correspond to pins PB30 to PB0, respectively. PBPRH and PBPRL always return the states of the pins regardless of the PFC setting.

(1) Port B Port Register H (PBPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB30 PR	PB29 PR	PB28 PR	PB27 PR	PB26 PR	PB25 PR	PB24 PR	PB23 PR	PB22 PR	PB21 PR	PB20 PR	PB19 PR	PB18 PR	PB17 PR	PB16 PR
Initial value:	0	PB30	PB29	PB28	PB27	PB26	PB25	PB24	PB23	PB22	PB21	PB20	PB19	PB18	PB17	PB16
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	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14	PB30PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
13	PB29PR	Pin state	R	
12	PB28PR	Pin state	R	
11	PB27PR	Pin state	R	
10	PB26PR	Pin state	R	
9	PB25PR	Pin state	R	
8	PB24PR	Pin state	R	
7	PB23PR	Pin state	R	
6	PB22PR	Pin state	R	
5	PB21PR	Pin state	R	
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
	PB15 PR	PB14 PR	PB13 PR	PB12 PR	PB11 PR	PB10 PR	PB9 PR	PB8 PR	PB7 PR	PB6 PR	PB5 PR	PB4 PR	PB3 PR	PB2 PR	PB1 PR	PB0 PR
Initial value:	PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15	PB15PR	Pin state	R	The pin state is returned regardless of the PFC setting. These bits cannot be modified.
14	PB14PR	Pin state	R	
13	PB13PR	Pin state	R	
12	PB12PR	Pin state	R	
11	PB11PR	Pin state	R	
10	PB10PR	Pin state	R	
9	PB9PR	Pin state	R	
8	PB8PR	Pin state	R	
7	PB7PR	Pin state	R	
6	PB6PR	Pin state	R	
5	PB5PR	Pin state	R	
4	PB4PR	Pin state	R	
3	PB3PR	Pin state	R	
2	PB2PR	Pin state	R	
1	PB1PR	Pin state	R	
0	PB0PR	Pin state	R	

20.3 Port D

Port D is an input/output port with the 16 pins shown in figure 20.3.

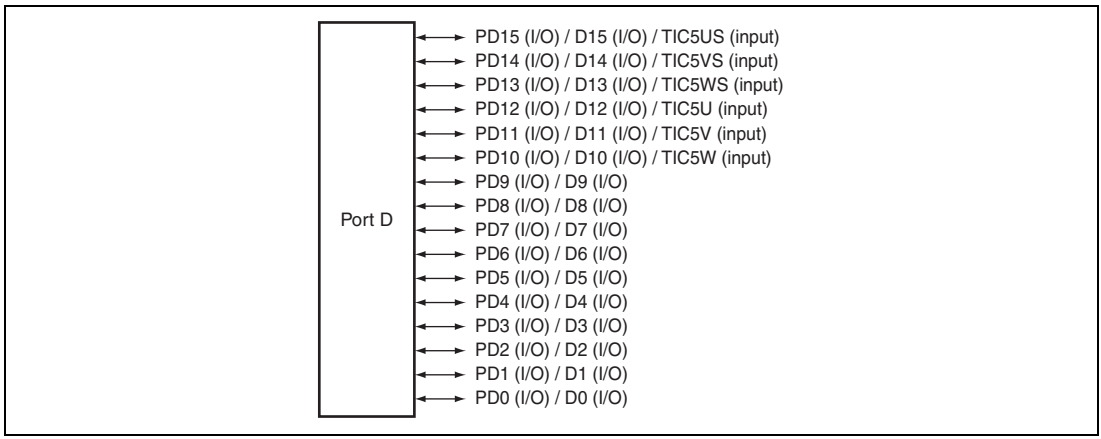


Figure 20.3 Port D

20.3.1 Register Descriptions

Table 20.5 lists the port D registers.

Table 20.5 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'xxxx	H'FFFE399E	8, 16

20.3.2 Port D Data Register L (PDDRL)

PDDRL is a 16-bit readable/writable register that stores port D data. Bits PD15DR to PD0DR correspond to pins PD15 to PD0, respectively.

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 20.8 summarizes PDDRL read/write operations.

PDDRL is initialized to the respective values shown in table 20.5 by a power-on reset, but is not initialized by a manual reset or in sleep mode or software standby mode.

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 20.6.
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 20.6 Port D Data Register L (PDDRL) Read/Write Operations

- PDDRL bits 15 to 0

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.

20.3.3 Port D Port Register L (PDPRL)

PDPRL is a 16-bit read-only register, in which bits PD15PR to PD0PR correspond to pins PD15 to PD0, respectively. PDPRL 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:	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
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	

20.4 Port F

Port F is an input/output port with the two pins shown in figure 20.4.

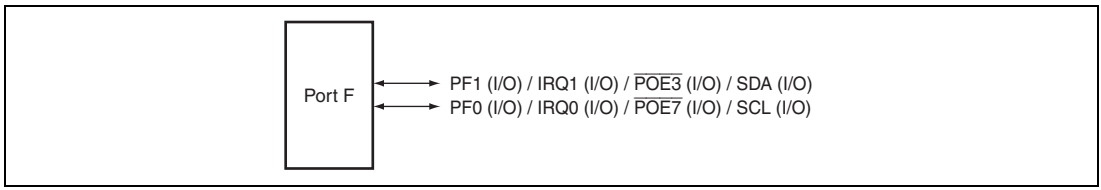


Figure 20.4 Port F

20.4.1 Register Descriptions

Table 20.7 lists the port F register.

Table 20.7 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R	H'0000	H'FFFE3A82	8, 16

20.4.2 Port F Data Register (PFDR)

PFDR is a 16-bit read-only register that stores port F data. Bits PF1DR and PF0DR correspond to pins PF1 and PF0, respectively.

Even if a value is written to PFDR, the value is not written into PFDR, and it does not affect the pin state. If PFDR is read, the pin state, not the register value, is returned directly. Table 20.8 summarizes PFDR read/write operations.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PF1DR	PF0DR
Initial value:	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: * Depends on the external pin state.

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	PF1DR	Pin state	R	See table 20.8.
0	PF0DR	Pin state	R	

Table 20.8 Port F Data Register (PFDR) Read/Write Operations

- PFDR bits 1 and 0

Pin Function	Read	Write
General input	Pin state	Ignored (no effect on pin state)
Other than general input	Pin state	Ignored (no effect on pin state)

20.5 Usage Note

When the PFC selects the following pin functions, the pin state cannot be read by accessing data registers or port registers.

- A25 to A0 (address bus)
- D15 to D0 (data bus)
- $\overline{\text{BS}}$
- $\overline{\text{CS7}}$ to $\overline{\text{CS0}}$
- RD/$\overline{\text{WR}}$
- $\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$ and $\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$
- $\overline{\text{RASL}}$ and $\overline{\text{CASL}}$
- CKE
- $\overline{\text{WAIT}}$
- $\overline{\text{BREQ}}$
- $\overline{\text{BACK}}$
- $\overline{\text{MRES}}$

Section 21 Flash Memory

This LSI has 512-kbyte on-chip flash memory. The flash memory has the following features.

21.1 Features

- Two flash-memory MATs, with one selected by the mode in which the LSI starts up

  The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting when the LSI starts up determines the memory MAT that is currently mapped. The MAT can be switched by bank-switching after the LSI has started up.

  — Size of the user MAT, from which booting-up proceeds after a power-on reset in user mode: 512 kbytes

  — Size of the user boot MAT, from which booting-up proceeds after a power-on reset in user boot mode: 12 kbytes

- Three on-board programming modes and one off-board programming mode

  On-board programming modes

  Boot Mode: The on-chip SCIF interface is used for programming in this mode. Either the user MAT or user-boot MAT can be programmed, and the bit rate for data transfer between the host and this LSI are automatically adjusted.

  User Program Mode: This mode allows programming of the user MAT via any desired interface.

  User Boot Mode: This mode allows writing of a user boot program via any desired interface and programming of the user MAT.

  Off-board programming mode

  Programmer Mode: This mode allows programming of the user MAT and user boot MAT with the aid of a PROM programmer.

- Downloading of an on-chip program to provide an interface for programming/erasure

  This LSI has a dedicated programming/erasing program. After this program has been downloaded to the on-chip RAM, programming or erasing can be performed by setting parameters as arguments. “User branching” is also supported.

— User branching

Programming is performed in 256-byte units. Each round of programming consists of application of the programming pulse, reading for verification, and several other steps. Erasing is performed in block units and each round of erasing consists of several steps. A user-processing routine can be executed between each round of erasing, and making the setting for this is called the addition of a user branch.

- Protection modes

  There are two modes of protection: software protection is applied by register settings and hardware protection is applied by the level on the FWE pin. Protection of the flash memory from programming or erasure can be selected.

  When an abnormal state is detected, such as runaway execution of programming/erasing, the protection modes initiate the transition to the error protection state and suspend programming/erasing processing.

- Programming/erasing time

  The time taken to program 256 bytes of flash memory in a single round is 2 ms (typ.), which is equivalent to 7.8 μs per byte. The erasing time is 80 ms (typ.) per 8-Kbyte block, 600 ms (typ.) per 64-Kbyte block, and 1200 ms (typ.) per 128-Kbyte block.

- Number of programming operations

  The flash memory can be programmed up to 100 times.

- Operating frequency for programming/erasing

  The operating frequency range for programming/erasing Iϕ = 32 to 40 MHz

21.2 Overview

21.2.1 Block Diagram

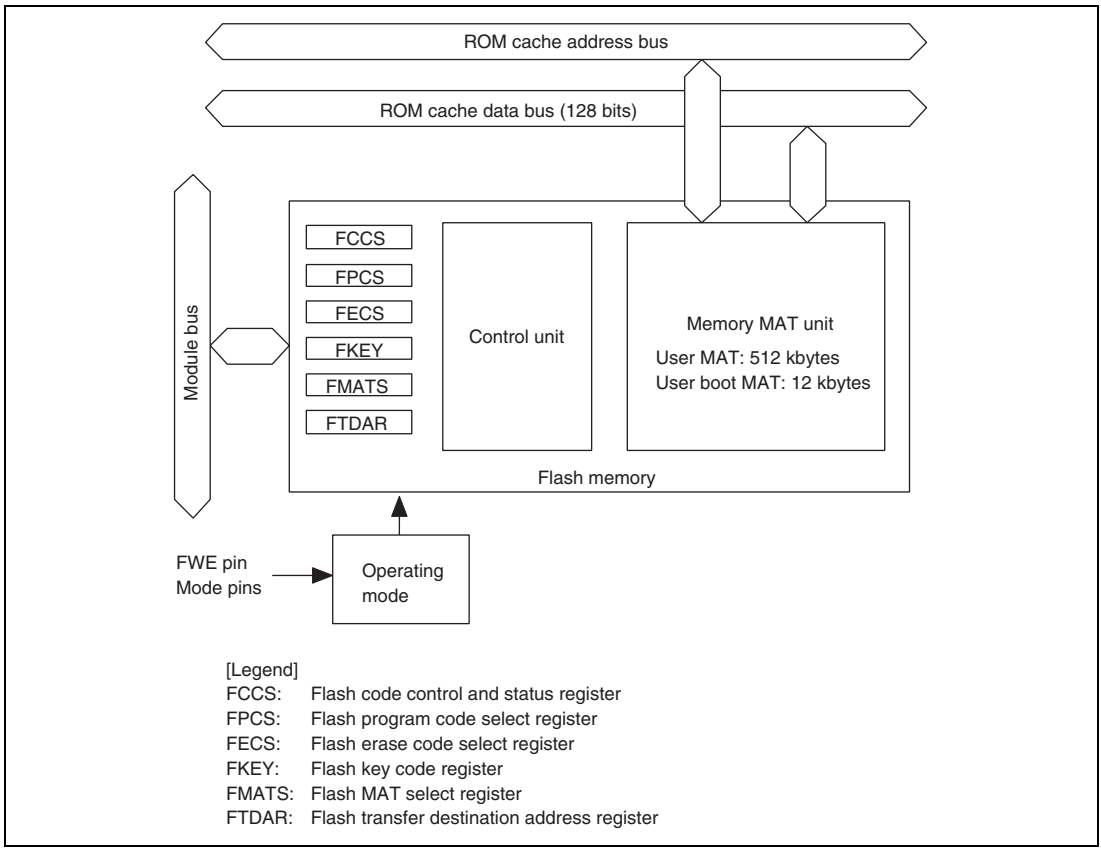


Figure 21.1 Block Diagram of Flash Memory

21.2.2 Operating Mode

When each mode pin and the FWE pin are set in the reset state and the reset signal is released, the microcontroller enters each operating mode as shown in figure 21.2. For the setting of each mode pin and the FWE pin, see table 21.1.

- Flash memory cannot be read, programmed, or erased in ROM invalid mode. The programming/erasing interface registers cannot be written to. When these registers are read, H'00 is always read.
- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

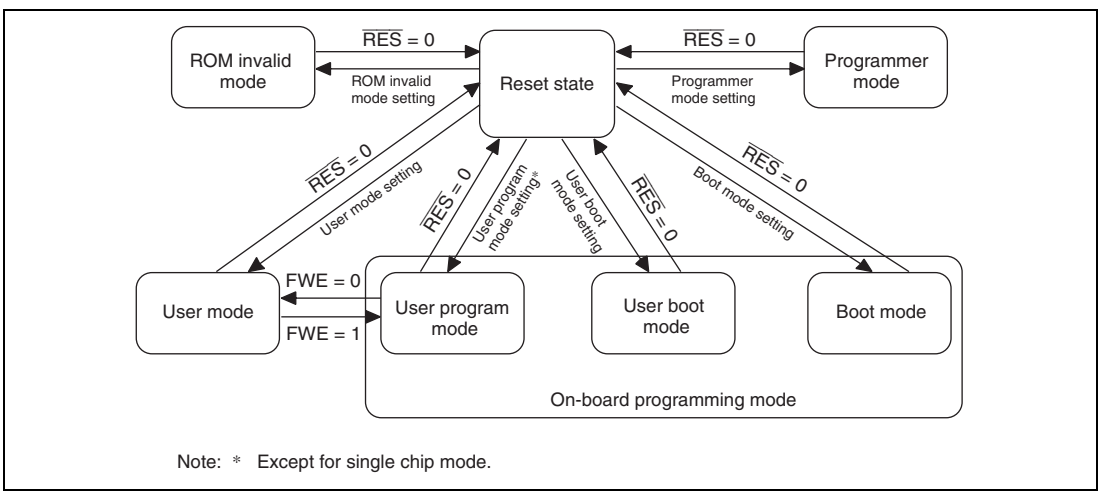


Figure 21.2 Mode Transition of Flash Memory

Table 21.1 Relationship between FWE and MD Pins and Operating Modes

Pin	Reset State	ROM Invalid Mode	User Mode	User Program Mode	User Boot Mode	Boot Mode	Programmer Mode
$\overline{\text{RES}}$	0	1	1	1	1	1	Setting value depends on the condition of the specialized PROM programmer.
FWE	0/1	0	0	1	1	1	
MD0	0/1	0/1*[1]	0/1*[2]	0	1	0	
MD1	0/1	0	1	1	0	0	

Notes: 1. MD0 = 0: 16-bit external bus, MD0 = 1: 8-bit external bus
2. MD0 = 0: External bus can be used, MD0 = 1: Single-chip mode (external bus cannot be used)

21.2.3 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 21.2.

Table 21.2 Comparison of Programming Modes

	Boot Mode	User Program Mode	User Boot Mode	Programmer Mode
Programming/ erasing environment	On-board programming	On-board programming	On-board programming	Off-board programming
Programming/ erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot MAT
Programming/ erasing control	Command method	Programming/ erasing interface	Programming/ erasing interface	—
All erasure	Possible (Automatic)	Possible	Possible	Possible (Automatic)
Block division erasure	Possible*1	Possible	Possible	Not possible
Program data transfer	From host via SCIF	From optional device via RAM	From optional device via RAM	Via programmer
User branch function	Not possible	Possible	Possible	Not possible
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	Embedded program storage MAT
Transition to user mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. Initiation starts from the embedded program storage MAT. After checking the flash-memory related registers, initiation starts from the reset vector of the user MAT.

- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- The user MAT and user boot MAT are all erased in boot mode. Then, the user MAT and user boot MAT can be programmed by means of the command method. However, the contents of the MAT cannot be read until this state.
  Only user boot MAT is programmed and the user MAT is programmed in user boot mode or only user MAT is programmed because user boot mode is not used.
- In user boot mode, the boot operation of the optional interface can be performed by a mode pin setting different from user program mode.

21.2.4 Flash Memory Configuration

This LSI's flash memory is configured by the 512-kbyte user MAT and 12-kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when the program execution or data access is performed between the two MATs, the MAT must be switched by using FMATS.

The user MAT or user boot MAT can be read in all modes if it is in ROM valid mode. However, the user boot MAT can be programmed only in boot mode and programmer mode.

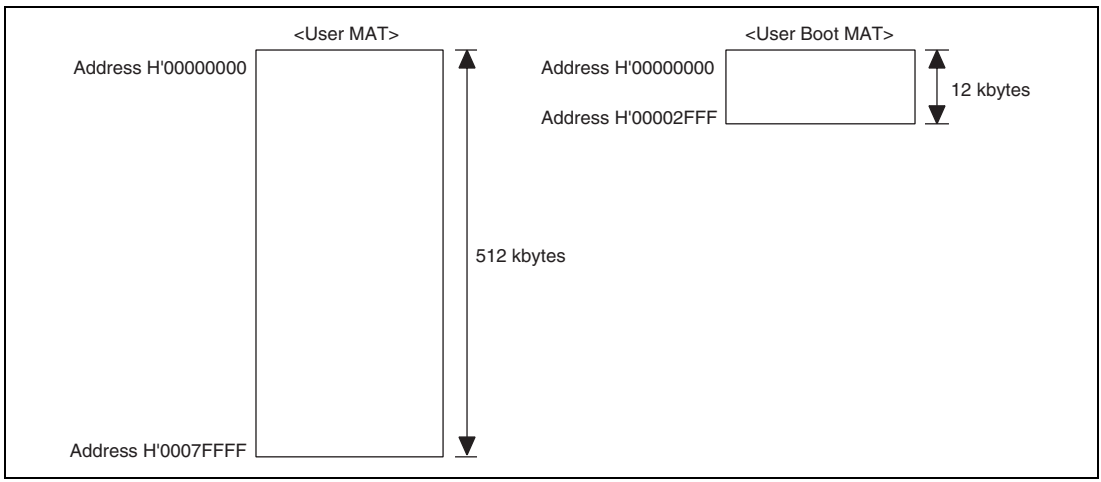


Figure 21.3 Flash Memory Configuration

The user MAT and user boot MAT have different memory sizes. Do not access a user boot MAT that is 12 kbytes or more. When a user boot MAT exceeding 12 kbytes is read from, an undefined value is read.

21.2.5 Block Division

The user MAT is divided into 128 kbytes (three blocks), 64 kbytes (one block), and 8 kbytes (eight blocks) as shown in figure 21.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB11 is specified when erasing.

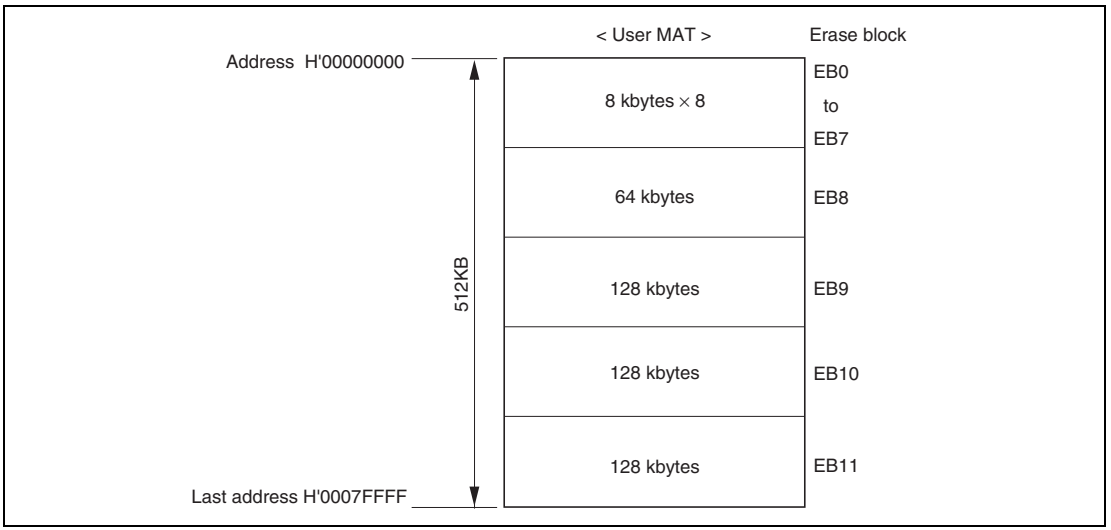


Figure 21.4 Block Division of User MAT

21.2.6 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface registers/parameters.

The procedure program is made by the user in user program mode and user boot mode. The overview of the procedure is as follows. For details, see section 21.5.2, User Program Mode.

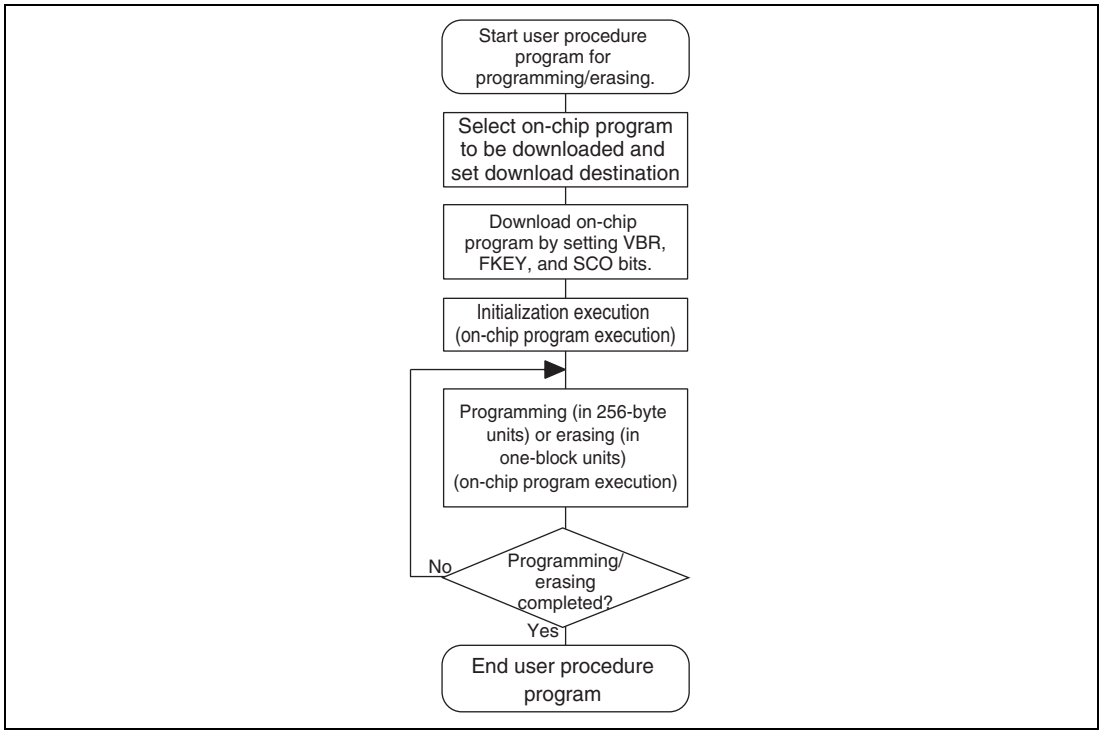


Figure 21.5 Overview of User Procedure Program

(1) Selection of On-Chip Program to be Downloaded and Setting of Download Destination

This LSI has programming/erasing programs and they can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface registers. The download destination can be specified by FTDAR.

(2) Download of On-Chip Program

The on-chip program is automatically downloaded by clearing VBR of the CPU to H'80000000 and then setting the SCO bit in the flash code control and status register (FCCS) and the flash key code register (FKEY), which are programming/erasing interface registers.

The user MAT is replaced to the embedded program storage area when downloading. Since the flash memory cannot be read when programming/erasing, the procedure program, which is working from download to completion of programming/erasing, must be executed in a space other than the flash memory to be programmed/erased (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameters, whether the normal download is executed or not can be confirmed.

Note that VBR can be changed after download is completed.

(3) Initialization of Programming/Erasing

The operating frequency and user branch are set before execution of programming/erasing. The user branch destination must be in an area other than the user MAT area which is in the middle of programming and the area where the on-chip program is downloaded. These settings are performed by using the programming/erasing interface parameters.

(4) Programming/Erasing Execution

To program or erase, the FWE pin must be brought high and user program mode must be entered.

The program data/programming destination address is specified in 256-byte units when programming.

The block to be erased is specified in erase-block units when erasing.

These specifications are set by using the programming/erasing interface parameters and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction to perform the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash memory.

There are limitations and notes on the interrupt processing during programming/erasing. For details, see section 21.7.2, Interrupts during Programming/Erasing.

(5) When Programming/Erasing is Executed Consecutively

When the processing is not ended by the 256-byte programming or one-block erasure, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

Since the downloaded on-chip program is left in the on-chip RAM after the processing, download and initialization are not required when the same processing is executed consecutively.

21.3 Input/Output Pins

Flash memory is controlled by the pins as shown in table 21.3.

Table 21.3 Pin Configuration

Pin Name	Symbol	Input/Output	Function
Power-on reset	$\overline{\text{RES}}$	Input	Reset
Flash programming enable	FWE	Input	Hardware protection when programming flash memory
Mode 1	MD1	Input	Sets operating mode of this LSI
Mode 0	MD0	Input	Sets operating mode of this LSI
Transmit data	TXD1 (PA25)	Output	Serial transmit data output (used in boot mode)
Receive data	RXD1 (PA24)	Input	Serial receive data input (used in boot mode)

21.4 Register Descriptions

21.4.1 Registers

The registers/parameters which control flash memory when the on-chip flash memory is valid are shown in table 21.4.

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence of operating modes and registers/parameters for use is shown in table 21.5.

Table 21.4 (1) Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash code control and status register	FCCS	R, W*[1]	H'00*[2] H'80*[2]	H'8000C000	8
Flash program code select register	FPCS	R/W	H'00	H'8000C001	8
Flash erase code select register	FECS	R/W	H'00	H'8000C002	8
Flash key code register	FKEY	R/W	H'00	H'8000C004	8
Flash MAT select register	FMATS	R/W	H'00*[3] H'AA*[3]	H'8000C005	8
Flash transfer destination address register	FTDAR	R/W	H'00	H'8000C006	8

Notes: 1. The bits except the SCO bit are read-only bits. The SCO bit is a programming-only bit. (The value which can be read is always 0.)
2. The initial value of the FWE bit is 0 when the FWE pin goes low. The initial value of the FWE bit is 1 when the FWE pin goes high.
3. The initial value at initiation in user mode or user program mode is H'00. The initial value at initiation in user boot mode is H'AA.

Table 21.4 (2) Parameter Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Download pass/fail result	DPFR	R/W	Undefined	On-chip RAM*	8, 16, 32
Flash pass/fail result	FPFR	R/W	Undefined	R0 of CPU	8, 16, 32
Flash multipurpose address area	FMPAR	R/W	Undefined	R5 of CPU	8, 16, 32
Flash multipurpose data destination area	FMPDR	R/W	Undefined	R4 of CPU	8, 16, 32
Flash erase block select	FEBS	R/W	Undefined	R4 of CPU	8, 16, 32
Flash program and erase frequency control	FPEFEQ	R/W	Undefined	R4 of CPU	8, 16, 32
Flash user branch address set parameter	FUBRA	R/W	Undefined	R5 of CPU	8, 16, 32

Note: * One byte of the start address in the on-chip RAM area specified by FTDAR is valid.

Table 21.5 Register/Parameter and Target Mode

		Download	Initialization	Programming	Erasure	Read	RAM Emulation
Programming/ erasing interface registers	FCCS	√	—	—	—	—	—
	FPCS	√	—	—	—	—	—
	FECS	√	—	—	—	—	—
	FKEY	√	—	√	√	—	—
	FMATS	—	—	√*1	√*1	√*2	—
	FTDAR	√	—	—	—	—	—
Programming/ erasing interface parameters	DPFR	√	—	—	—	—	—
	FPFR	—	√	√	√	—	—
	FPEFEQ	—	√	—	—	—	—
	FUBRA	—	√	—	—	—	—
	FMPAR	—	—	√	—	—	—
	FMPDR	—	—	√	—	—	—
	FEBS	—	—	—	√	—	—

Notes: 1. The setting is required when programming or erasing user MAT in user boot mode.
2. The setting may be required according to the combination of initiation mode and read target MAT.

21.4.2 Programming/Erasing Interface Registers

The programming/erasing interface registers are as described below. They are all 8-bit registers that can be accessed in bytes.

(1) Flash Code Control and Status Register (FCCS)

FCCS is configured by bits which request the monitor of the FWE pin state and error occurrence during programming or erasing flash memory and the download of the on-chip program.

Bit:	7	6	5	4	3	2	1	0
	FWE	MAT	-	FLER	-	-	-	SCO
Initial value:	1/0	1/0	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	FWE	1/0	R	Flash Programming Enable Monitors the level which is input to the FWE pin that performs hardware protection of the flash memory programming or erasing. The initial value is 0 or 1 according to the FWE pin state. 0: When the FWE pin goes low (in hardware protection state) 1: When the FWE pin goes high
6	MAT	1/0	R	MAT Bit Indicates whether the user MAT or user boot MAT is selected. 0: User MAT is selected 1: User boot MAT is selected
5	—	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
4	FLER	0	R	Flash Memory Error Indicates an error occurs during programming and erasing flash memory. When FLER is set to 1, flash memory enters the error protection state. When FLER is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset signal must be released after the reset period of 100 μs which is longer than normal. 0: Flash memory operates normally Programming/erasing protection for flash memory (error protection) is invalid. [Clearing condition] At a power-on reset 1: Indicates an error occurs during programming/erasing flash memory. Programming/erasing protection for flash memory (error protection) is valid. [Setting condition] See section 21.6.3, Error Protection.
3 to 1	—	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
0	SCO	0	(R)/W	Source Program Copy Operation Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM. When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM area specified by FTDAR. In order to set this bit to 1, H'A5 must be written to FKEY and this operation must be in the on-chip RAM. Thirty-two NOP instructions must be executed immediately after setting this bit to 1. For interrupts during download, see section 21.7.2, Interrupts during Programming/Erasing. For the download time, see section 21.7.3, Other Notes. Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1. Download by setting the SCO bit to 1 requires a special interrupt processing that performs bank switching to the on-chip program storage area. Therefore, before issuing a download request (SCO = 1), set VBR to H'80000000. Otherwise, the CPU gets out of control. Once download end is confirmed, VBR can be changed to any other value. The mode in which the FWE pin is high must be used when using the SCO function. 0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed. [Clearing condition] When download is completed 1: Request that the on-chip programming/erasing program is downloaded to the on-chip RAM is generated [Setting conditions] When all of the following conditions are satisfied and 1 is written to this bit • FKEY is written to H'A5 • During execution in the on-chip RAM

(2) Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PPVS	0	R/W	Program Pulse Single Selects the programming program. 0: On-chip programming program is not selected [Clearing condition] When transfer is completed 1: On-chip programming program is selected

(3) Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program. 0: On-chip erasing program is not selected [Clearing condition] When transfer is completed 1: On-chip erasing program is selected

(4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of the on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download the on-chip program or executing the downloaded programming/erasing program, these processings cannot be executed if the key code is not written.

Bit:	7	6	5	4	3	2	1	0
	K[7:0]							
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 0	K[7:0]	All 0	R/W	Key Code Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be written to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed. Only when H'5A is written, programming/erasing of flash memory can be executed. Even if the on-chip programming/erasing program is executed, flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY. H'A5: Writing to the SCO bit is enabled (The SCO bit cannot be set by a value other than H'A5.) H'5A: Programming/erasing is enabled (A value other than H'5A enables software protection state.) H'00: Initial value

(5) Flash MAT Select Register (FMATS)

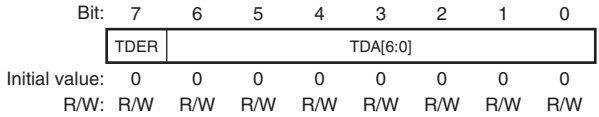
FMATS specifies whether user MAT or user boot MAT is selected.

Bit:	7	6	5	4	3	2	1	0
	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
Initial value:	0/1	0	0/1	0	0/1	0	0/1	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	MS7	0/1	R/W	MAT Select
6	MS6	0	R/W	These bits are in user-MAT selection state when a value other than H'AA is written and in user-boot-MAT selection state when H'AA is written. The MAT is switched by writing a value in FMATS with the on-chip RAM instruction. When the MAT is switched, follow section 21.7.1, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user program mode if user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or in programmer mode.) H'AA: The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H'AA) Initial value when these bits are initiated in user boot mode. H'00: Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state) [Programmable condition] These bits are in the execution state in the on-chip RAM.
5	MS5	0/1	R/W	
4	MS4	0	R/W	
3	MS3	0/1	R/W	
2	MS2	0	R/W	
1	MS1	0/1	R/W	
0	MS0	0	R/W	

(6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the on-chip RAM address to which the on-chip program is downloaded. Make settings for FTDAR before writing 1 to the SCO bit in FCCS. The initial value is H'00 which points to the start address (H'FFF81000) in on-chip RAM.



Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when there is an error in the download start address set by bits 6 to 0 (TDA6 to TDA0). Whether the address setting is erroneous or not is tested by checking whether the setting of TDA6 to TDA0 is in the range of H'00 to H'05 after setting the SCO bit in FCCS to 1 and performing download. Before setting the SCO bit to 1 be sure to set the FTDAR value between H'00 to H'05 as well as clearing this bit to 0. 0: Setting of TDA6 to TDA0 is normal 1: Setting of TDER and TDA6 to TDA0 is H'06 to H'FF and download has been aborted
6 to 0	TDA[6:0]	All 0	R/W	Transfer Destination Address These bits specify the download start address. A value from H'00 to H'05 can be set to specify the download start address in on-chip RAM in 2-kbyte units. A value from H'06 to H'7F cannot be set. If such a value is set, the TDER bit (bit 7) in this register is set to 1 to prevent download from being executed. H'00: Download start address is set to H'FFF81000 H'01: Download start address is set to H'FFF81800 H'02: Download start address is set to H'FFF82000 H'03: Download start address is set to H'FFF82800 H'04: Download start address is set to H'FFF83000 H'05: Download start address is set to H'FFF83800 H'06 to H'7F: Setting prohibited. If this value is set, the TDER bit (bit 7) is set to 1 to abort the download processing.

21.4.3 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, user branch destination address, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. This parameter uses the general registers of the CPU (R4, R5, and R0) or the on-chip RAM area. The initial value is undefined.

At download all CPU registers are stored, and at initialization or when the on-chip program is executed, CPU registers except for R0 are stored. The return value of the processing result is written in R0. Since the stack area is used for storing the registers or as a work area, the stack area must be saved at the processing start. (The maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters are used in the following four items.

1. Download control
2. Initialization before programming or erasing
3. Programming
4. Erasing

These items use different parameters. The correspondence table is shown in table 21.6.

The processing results of initialization, programming, and erasing are returned, but the bit contents have different meanings according to the processing program. See the description of FPFR for each processing.

Table 21.6 Usable Parameters and Target Modes

Name of Parameter	Abbreviation	Download	Initialization	Programming	Erasure	R/W	Initial Value	Allocation
Download pass/fail result	DPFR	√	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	√	√	√	R/W	Undefined	R0 of CPU
Flash programming/ erasing frequency control	FPEFEQ	—	√	—	—	R/W	Undefined	R4 of CPU
Flash user branch address set	FUBRA	—	√	—	—	R/W	Undefined	R5 of CPU
Flash multipurpose address area	FMPAR	—	—	√	—	R/W	Undefined	R5 of CPU
Flash multipurpose data destination area	FMPDR	—	—	√	—	R/W	Undefined	R4 of CPU
Flash erase block select	FEBS	—	—	—	√	R/W	Undefined	R4 of CPU

Note: * One byte of start address of download destination specified by FTDAR

(1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area to be downloaded is the area as much as 3 kbytes starting from the start address specified by FTDAR. For the address map of the on-chip RAM, see figure 21.10.

The download control is set by using the programming/erasing interface registers. The return value is given by the DPFR parameter.

(a) Download Pass/Fail Result Parameter (DPFR: One Byte of Start Address of On-Chip RAM Specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading is executed or not. Since the confirmation whether the SCO bit is set to 1 is difficult, the certain determination must be performed by setting one byte of the start address of the on-chip RAM area specified by FTDAR to a value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1). For

the checking method of download results, see section 21.5.2 (2), Programming Procedure in User Program Mode.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SS	FK	SF
Initial value:	-	-	-	-	-	-	-	-
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 3	—	Undefined	R/W	Unused Return 0.
2	SS	Undefined	R/W	Source Select Error Detect The on-chip program which can be downloaded can be specified as only one type. When more than two types of the program are selected, the program is not selected, or the program is selected without mapping, an error occurs. 0: Download program can be selected normally 1: Download error occurs (Multi-selection or program which is not mapped is selected)
1	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result whether the value of FKEY is set to H'A5. 0: FKEY setting is normal (FKEY = H'A5) 1: FKEY setting is abnormal (FKEY = value other than H'A5)
0	SF	Undefined	R/W	Success/Fail Returns the result whether download has ended normally or not. 0: Downloading on-chip program has ended normally (no error) 1: Downloading on-chip program has ended abnormally (error occurs)

(2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

The specified period pulse must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. The operating frequency of the CPU must be set. Since the user branch function is supported, the user branch destination address must be set.

The initial program is set as a parameter of the programming/erasing program which has downloaded these settings.

(2.1) Flash Programming/Erasing Frequency Parameter (FPEFEQ: General Register R4 of CPU)

This parameter sets the operating frequency of the CPU.

The flash programming/erasing frequency Iϕ of this LSI is limited to 32 to 40 MHz.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	Undefined	R/W	Unused Return 0.
15 to 0	F15 to F0	Undefined	R/W	Frequency Set Set the operating frequency Iϕ of the CPU following the calculation below. $I\phi = F[15:0] \times 10^4$ Hz 1. Round it off to the digit of 1 kHz, and round down the lower digits. 2. For example, when Iϕ = 33.333 MHz, set as follows: (1) $I\phi = 3333 \times 10^4$ Hz (2) F[15:0] = 3333 (H'0D05) (3) Set R4 (FPEFEQ) to H'00000D05.

(2.2) Flash User Branch Address Setting Parameter (FUBRA: General Register R5 of CPU)

This parameter sets the user branch destination address. The user program which has been set can be executed in specified processing units when programming and erasing.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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	UA31 to UA0	Undefined	R/W	User Branch Destination Address When the user branch is not required, address 0 (H'00000000) must be set. The user branch destination must be an area other than the flash memory, an area other than the RAM area in which on-chip program has been transferred, or the external bus space. Note that the CPU must not branch to an area without the execution code and get out of control. The on-chip program download area and stack area must not be overwritten. If CPU runaway occurs or the download area or stack area is overwritten, the value of flash memory cannot be guaranteed. The download of the on-chip program, initialization, initiation of the programming/erasing program must not be executed in the processing of the user branch destination. Programming or erasing cannot be guaranteed when returning from the user branch destination. The program data which has already been prepared must not be programmed. Store general registers R8 to R15. General registers R0 to R7 are available without storing them. Moreover, the programming/erasing interface registers must not be written to in the processing of the user branch destination. After the processing of the user branch has ended, the programming/erasing program must be returned to by using the RTS instruction. For the execution intervals of the user branch processing, see note 2 (User branch processing intervals) in section 21.7.3, Other Notes.

(2.3) Flash Pass/Fail Result Parameter (FPFR: General Register R0 of CPU)

This parameter indicates the return value of the initialization result.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	-	-	-	-	-	-	-	-	-	-	-	-	-	BR	FQ	SF
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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 3	—	Undefined	R/W	Unused Return 0.
2	BR	Undefined	R/W	User Branch Error Detect Returns the check result whether the specified user branch destination address is in the area other than the storage area of the programming/erasing program which has been downloaded. 0: User branch address setting is normal 1: User branch address setting is abnormal
1	FQ	Undefined	R/W	Frequency Error Detect Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	Undefined	R/W	Success/Fail Indicates whether initialization is completed normally. 0: Initialization has ended normally (no error) 1: Initialization has ended abnormally (error occurs)

(3) Programming Execution

When flash memory is programmed, the programming destination address and programming data on the user MAT must be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT is set in general register R5 of the CPU. This parameter is called FMPAR (flash multipurpose address area parameter). Since the program data is always in 256-byte units, the lower eight bits (MOA7 to MOA0) must be H'00 as the boundary of the programming start address on the user MAT.
2. The program data for the user MAT must be prepared in the consecutive area. The program data must be in the consecutive space which can be accessed by using the MOV.B instruction of the CPU and is not the flash memory space.
   When data to be programmed does not satisfy 256 bytes, the 256-byte program data must be prepared by embedding the dummy code (H'FF).
   The start address of the area in which the prepared program data is stored must be set in general register R4. This parameter is called FMPDR (flash multipurpose data destination area parameter).

For details on the programming procedure, see section 21.5.2, User Program Mode.

(3.1) Flash Multipurpose Address Area Parameter (FMPAR: General Register R5 of CPU)

This parameter indicates the start address of the programming destination on the user MAT.

When an address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 256-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in FPFR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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	MOA31 to MOA0	Undefined	R/W	MOA31 to MOA0 Store the start address of the programming destination on the user MAT. The consecutive 256-byte programming is executed starting from the specified start address of the user MAT. The MOA7 to MOA0 bits are always 0 because the start address of the programming destination is at the 256-byte boundary.

(3.2) Flash Multipurpose Data Destination Area Parameter (FMPDR: General Register R4 of CPU)

This parameter indicates the start address in the area which stores the data to be programmed in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit (bit 2) in FPFR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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	MOD31 to MOD0	Undefined	R/W	MOD31 to MOD0 Store the start address of the area which stores the program data for the user MAT. The consecutive 256-byte data is programmed to the user MAT starting from the specified start address.

(3.3) Flash Pass/Fail Result Parameter (FPFR: General Register R0 of CPU)

This parameter indicates the return value of the program processing result.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	-	-	-	-	-	-	-	-	-	MD	EE	FK	-	WD	WA	SF
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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 7	—	Undefined	R/W	Unused Return 0.
6	MD	Undefined	R/W	Programming Mode Related Setting Error Detect Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is not entered. When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 21.6.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: FWE = 0 or FLER = 1, and programming cannot be performed

Bit	Bit Name	Initial Value	R/W	Description
5	EE	Undefined	R/W	Programming Execution Error Detect 1 is returned to this bit when the specified data could not be written because the user MAT was not erased or when flash-memory related register settings are partially changed on returning from the user branch processing. If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten. Programming of the user boot MAT must be executed in boot mode or programmer mode. 0: Programming has ended normally 1: Programming has ended abnormally (programming result is not guaranteed)
4	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY before the start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other than H'5A)
3	—	Undefined	R/W	Unused Return 0.
2	WD	Undefined	R/W	Write Data Address Error Detect When an address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs. 0: Setting of write data address is normal 1: Setting of write data address is abnormal

Bit	Bit Name	Initial Value	R/W	Description
1	WA	Undefined	R/W	Write Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • The programming destination address is an area other than flash memory • The specified address is not at the 256-byte boundary (A7 to A0 are not 0) 0: Setting of programming destination address is normal 1: Setting of programming destination address is abnormal
0	SF	Undefined	R/W	Success/Fail Indicates whether the program processing has ended normally or not. 0: Programming has ended normally (no error) 1: Programming has ended abnormally (error occurs)

(4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program which is downloaded. This is set to the FEBS parameter (general register R4).

One block is specified from the block number 0 to 15.

For details on the erasing procedure, see section 21.5.2, User Program Mode.

(4.1) Flash Erase Block Select Parameter (FEBS: General Register R4 of CPU)

This parameter specifies the erase-block number. Several block numbers cannot be specified.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	-	-	-	-	-	-	-	-	EBS[7:0]							
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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 8	—	Undefined	R/W	Unused Return 0.
7 to 0	EBS[7:0]	Undefined	R/W	Set the erase-block number in the range from 0 to 11. 0 corresponds to the EB0 block and 11 corresponds to the EB11 block. An error occurs when a number other than 0 to 11 (H'00 to H'0B) is set.

(4.2) Flash Pass/Fail Result Parameter (FPFR: General Register R0 of CPU)

This parameter returns the value of the erasing processing result.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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
	-	-	-	-	-	-	-	-	-	MD	EE	FK	EB	-	-	SF
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	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 7	—	Undefined	R/W	Unused Return 0.
6	MD	Undefined	R/W	Erasure Mode Related Setting Error Detect Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is not entered. When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 21.6.3, Error Protection. 0: FWE and FLER settings are normal (FWE = 1, FLER = 0) 1: FWE = 0 or FLER = 1, and erasure cannot be performed

Bit	Bit Name	Initial Value	R/W	Description
5	EE	Undefined	R/W	Erasure Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed on returning from the user branch processing. If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT. If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased. Erasure of the user boot MAT must be executed in boot mode or programmer mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure result is not guaranteed)
4	FK	Undefined	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before start of the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other than H'5A)
3	EB	Undefined	R/W	Erase Block Select Error Detect Returns the check result whether the specified erase-block number is in the block range of the user MAT. 0: Setting of erase-block number is normal 1: Setting of erase-block number is abnormal
2, 1	—	Undefined	R/W	Unused Return 0.
0	SF	Undefined	R/W	Success/Fail Indicates whether the erasing processing has ended normally or not. 0: Erasure has ended normally (no error) 1: Erasure has ended abnormally (error occurs)

21.5 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the on-board programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: user program mode, user boot mode, and boot mode.

For details on the pin setting for entering each mode, see table 21.1. For details on the state transition of each mode for flash memory, see figure 21.2.

21.5.1 Boot Mode

Boot mode executes programming/erasing user MAT and user boot MAT by means of the control command and program data transmitted from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pin is set in boot mode, the boot program in the microcontroller is initiated. After the SCI bit rate is automatically adjusted, the communication with the host is executed by means of the control command method.

The system configuration diagram in boot mode is shown in figure 21.6. For details on the pin setting in boot mode, see table 21.1. Interrupts are ignored in boot mode, so do not generate them. Note that the AUD cannot be used during boot mode operation.

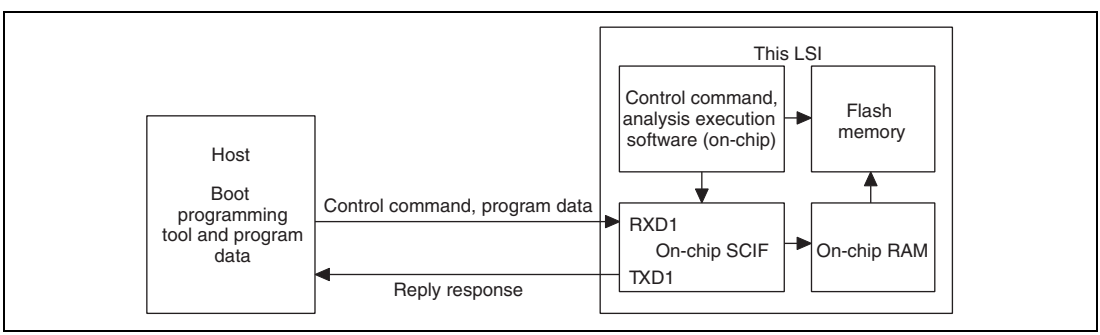


Figure 21.6 System Configuration in Boot Mode

(1) SCIF Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCIF-communication data (H'00), which is transmitted consecutively by the host. The SCIF transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and transmits 1 byte of H'55 to this LSI. When reception is not executed normally, boot mode is initiated again (reset) and the operation described above must be executed. The bit rate between the host and this LSI is not matched because of the bit rate of transmission by the host and system clock frequency of this LSI. To operate the SCIF normally, the transfer bit rate of the host must be set to 9,600 bps or 19,200 bps.

The system clock frequency which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI is shown in table 21.7. Boot mode must be initiated in the range of this system clock.

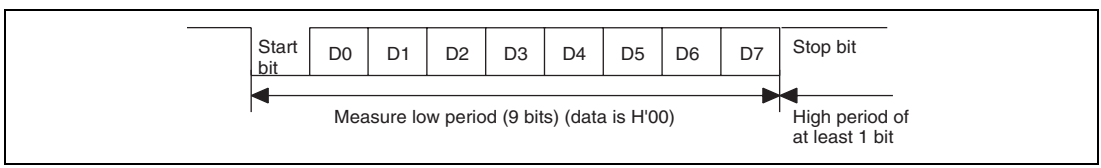


Figure 21.7 Automatic Adjustment Operation of SCIF Bit Rate

Table 21.7 Peripheral Clock (Pϕ) Frequency that Can Automatically Adjust Bit Rate of This LSI

Host Bit Rate	Peripheral Clock (Pϕ) Frequency That Can Automatically Adjust LSI's Bit Rate
9,600 bps	32 to 40 MHz
19,200 bps	32 to 40 MHz

(2) State Transition Diagram

Figure 21.8 gives an overview of the state transitions after the chip has been started up in boot mode. For details on boot mode, see section 21.8.1, Specifications of the Standard Serial Communications Interface in Boot Mode.

1. Bit-rate matching

   After the chip has been started up in boot mode, bit-rate matching between the SCI and the host proceeds.
2. Waiting for inquiry and selection commands

   The chip sends the requested information to the host in response to inquiries regarding the size and configuration of the user MAT, start addresses of the MATs, information on supported devices, etc.
3. Automatic erasure of the entire user MAT and user boot MAT

   After all necessary inquiries and selections have been made and the command for transition to the programming/erasure state is sent by the host, the entire user MAT and user boot MAT are automatically erased.
4. Waiting for programming/erasure command
   - On receiving the programming selection command, the chip waits for data to be programmed. To program data, the host transmits the programming command code followed by the address where programming should start and the data to be programmed. This is repeated as required while the chip is in the programming-selected state. To terminate programming, H'FFFFFFFF should be transmitted as the first address of the area for programming. This makes the chip return to the programming/erasure command waiting state from the programming data waiting state.
   - On receiving the erasure select command, the chip waits for the block number of a block to be erased. To erase a block, the host transmits the erasure command code followed by the number of the block to be erased. This is repeated as required while the chip is in the erasure-selected state. To terminate erasure, H'FF should be transmitted as the block number. This makes the chip return to the programming/erasure command waiting state from the erasure block number waiting state. Erasure should only be executed when a specific block is to be reprogrammed without executing a reset-start of the chip after the flash memory has been programmed in boot mode. If all desired programming is done in a single operation, such erasure processing is not necessary because all blocks are erased before the chip enters the programming/erasure/other command waiting state.
   - In addition to the programming and erasure commands, commands for sum checking and blank checking (checking for erasure) of the user MAT and user boot MAT, reading data from the user MAT/user boot MAT, and acquiring current state information are provided.

Note that the command for reading from the user MAT/user boot MAT can only read data that has been programmed after automatic erasure of the entire user MAT and user boot MAT.

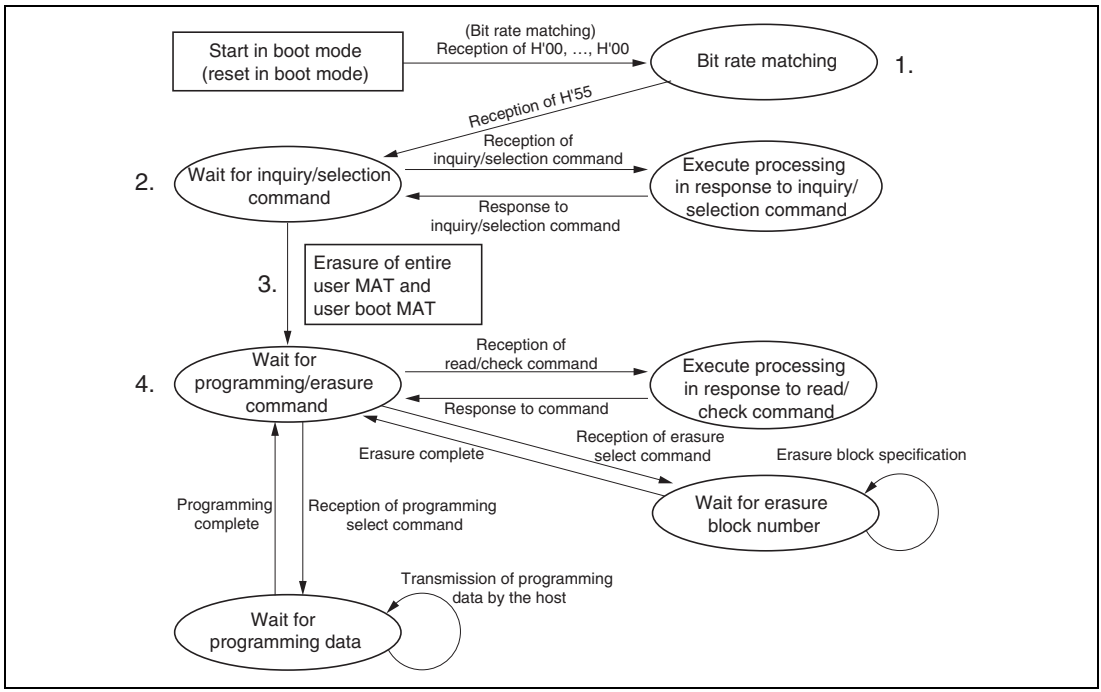


Figure 21.8 State Transitions in Boot Mode

21.5.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

Programming/erasing is executed by downloading the program in the microcontroller.

The overview flow is shown in figure 21.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, transition to reset must not be executed. Doing so may cause damage or destroy flash memory. If reset is executed accidentally, the reset signal must be released after the reset input period, which is longer than the normal 100 μs.

For details on the programming procedure, see the description in section 21.5.2 (2), Programming Procedure in User Program Mode. For details on the erasing procedure, see the description in section 21.5.2 (3), Erasing Procedure in User Program Mode.

For the overview of a processing that repeats erasing and programming by downloading the programming program and the erasing program in separate on-chip ROM areas using FTDAR, see the description in section 21.5.2 (4), Erasing and Programming Procedure in User Program Mode.

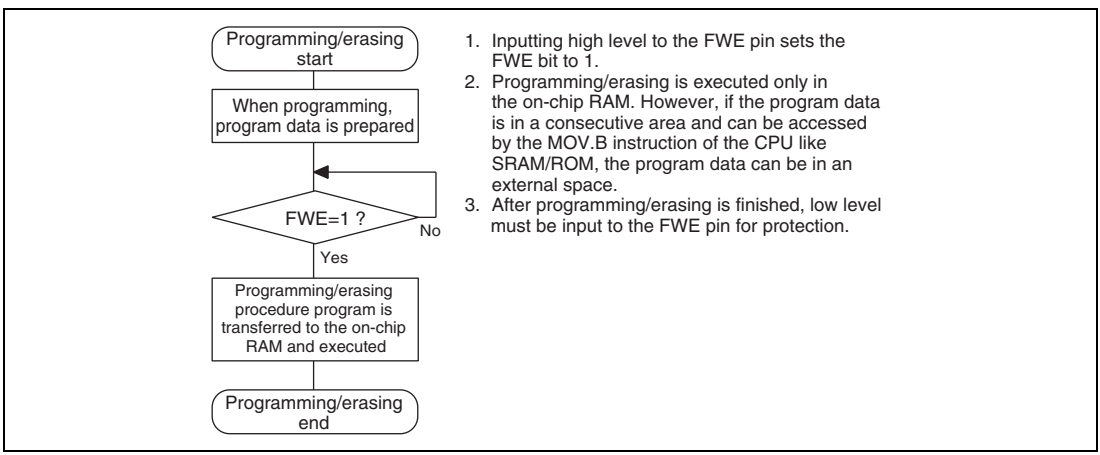


Figure 21.9 Programming/Erasing Overview Flow

(1) On-Chip RAM Address Map when Programming/Erasing is Executed

Parts of the procedure program that are made by the user, like download request, programming/erasing procedure, and decision of the result, must be executed in the on-chip RAM. All of the on-chip program that is to be downloaded is in on-chip RAM. Note that on-chip RAM must be controlled so that these parts do not overlap.

Figure 21.10 shows the program area to be downloaded.

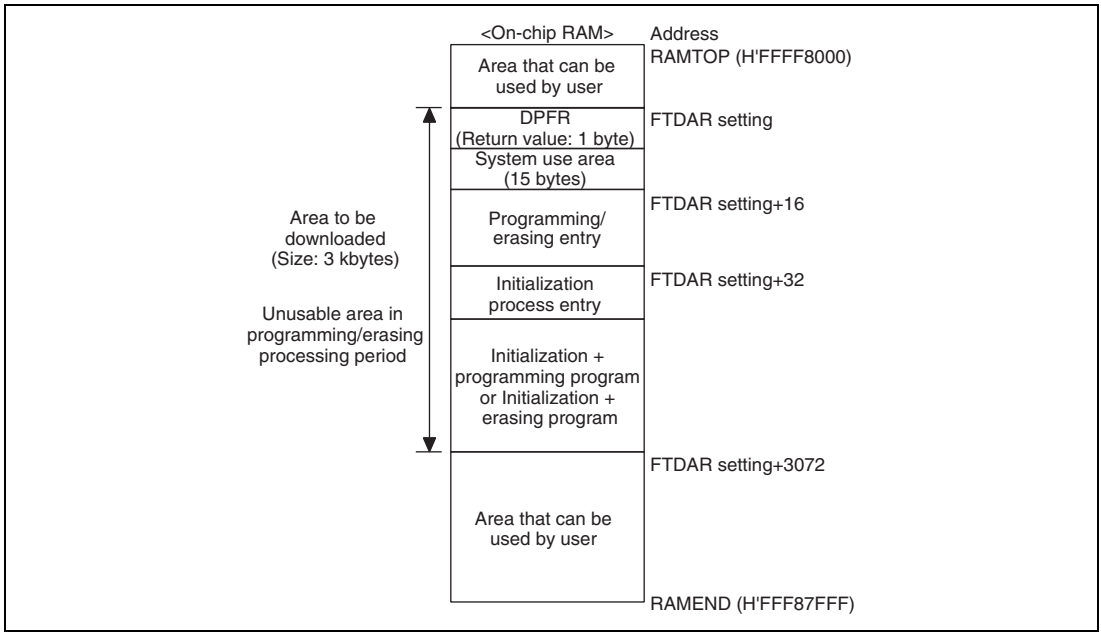


Figure 21.10 RAM Map after Download

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 21.11.

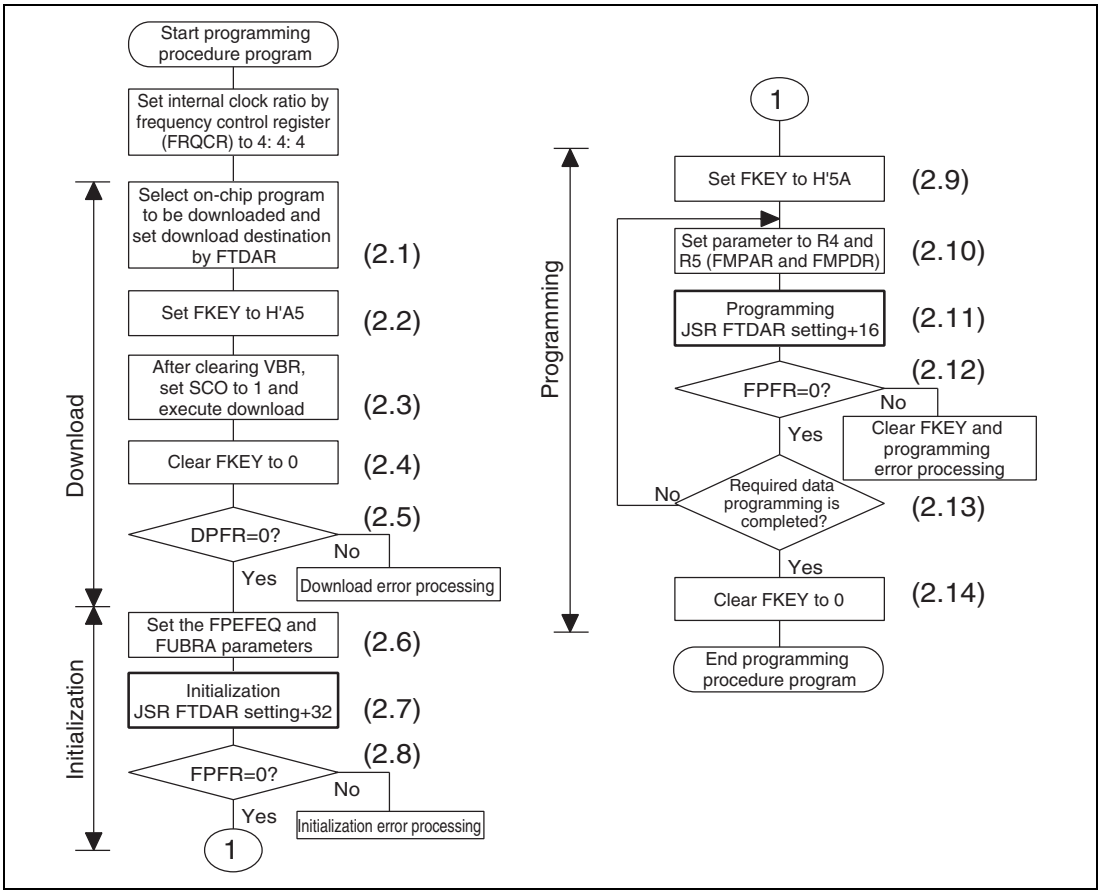


Figure 21.11 Programming Procedure

The details of the programming procedure are described below. The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM. Specify 4:4:4 as the frequency division ratios of an internal clock (Iϕ), a bus clock (Bϕ), and a peripheral clock (Pϕ) through the frequency control register (FRQCR).

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedural Program and Data for Programming.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been executed, carry out erasing before writing.

256-byte programming is performed in one program processing. When more than 256-byte programming is performed, programming destination address/program data parameter is updated in 256-byte units and programming is repeated.

When less than 256-byte programming is performed, data must total 256 bytes by adding the invalid data. If the invalid data to be added is H'FF, the program processing period can be shortened.

(2.1) Select the on-chip program to be downloaded

When the PPVS bit of FPCS is set to 1, the programming program is selected.
Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.
Specify the start address of the download destination by FTDAR.

(2.2) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for a download request.

(2.3) VBR is set to 0 and 1 is written to the SCO bit of FCCS, and then download is executed.

VBR must always be set to H'80000000 before setting the SCO bit to 1.
To write 1 to the SCO bit, the following conditions must be satisfied.

- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.
The download result can be confirmed only by the return value of the DPFR parameter. Before the SCO bit is set to 1, incorrect decision must be prevented by setting the DPFR parameter, that is one byte of the start address of the on-chip RAM area specified by FTDAR, to a value other than the return value (H'FF).
When download is executed, particular interrupt processing, which is accompanied by the bank switch as described below, is performed as an internal microcontroller processing, so VBR need to be set to H'80000000. Thirty-two NOP instructions are executed immediately after the instructions that set the SCO bit to 1.

1. The user MAT space is switched to the on-chip program storage area.
2. After the selection condition of the download program and the address set in FTDAR are checked, the transfer processing is executed starting to the on-chip RAM address specified by FTDAR.
3. The SCO bits in FCCS, FPCS, and FECS are cleared to 0.
4. The return value is set to the DPFR parameter.
5. After the on-chip program storage area is returned to the user MAT space, execution returns to the user procedure program.

After download is completed and the user procedure program is running, the VBR setting can be changed.

The notes on download are as follows.

In the download processing, the values of the general registers of the CPU are retained.

During the download processing, interrupts must not be generated. For details on the relationship between download and interrupts, see section 21.7.2, Interrupts during Programming/Erasing.

Since a stack area of maximum 256 bytes is used, an area of at least 128 bytes must be saved before setting the SCO bit to 1.

If flash memory is accessed by the DMAC during downloading, operation cannot be guaranteed. Therefore, access by the DMAC must not be executed.

(2.4) FKEY is cleared to H'00 for protection.

(2.5) The value of the DPFR parameter must be checked to confirm the download result.

A recommended procedure for confirming the download result is shown below.

1. Check the value of the DPFR parameter (one byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
2. If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit (bit 7) in FTDAR.
3. If the value of the DPFR parameter is different from before downloading, check the SS bit (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download program selection and FKEY register setting were normal, respectively.

(2.6) The operating frequency is set to the FPEFEQ parameter and the user branch destination is set to the FUBRA parameter for initialization.

1. The current frequency of the CPU clock is set to the FPEFEQ parameter (general register R4). The settable range Iϕ of the FPEFEQ parameter is 32 to 40 MHz.
   When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in section 21.4.3 (2.1), Flash Programming/Erasing Frequency Parameter (FPEFEQ: General Register R4 of CPU).
2. The start address in the user branch destination is set to the (FUBRA: CPU general register R5) parameter.
   When the user branch processing is not required, 0 must be set to FUBRA.
   When the user branch is executed, the branch destination is executed in flash memory other than the one that is to be programmed. The area of the on-chip program that is downloaded cannot be set.
   The program processing must be returned from the user branch processing by the RTS instruction.
   See the description in section 21.4.3 (2.2), Flash User Branch Address Setting Parameter (FUBRA: General Register R5 of CPU).

(2.7) Initialization

When a programming program is downloaded, the initialization program is also downloaded to on-chip RAM. There is an entry point of the initialization program in the area from (download start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

```
MOV.L  #DLTOP+32,R1          ; Set entry address to R1
JSR    @R1                   ; Call initialization routine
NOP
```

1. The general registers other than R0 are saved in the initialization program.
2. R0 is a return value of the FPFR parameter.
3. Since the stack area is used in the initialization program, a stack area of 256 bytes or more must be reserved in RAM.
4. Interrupts can be accepted during the execution of the initialization program. However, the program storage area and stack area in on-chip RAM and register values must not be destroyed.

(2.8) The return value of the initialization program, FPFR (general register R0) is checked.

(2.9) FKEY must be set to H'5A and the user MAT must be prepared for programming.

(2.10) The parameter which is required for programming is set.

The start address of the programming destination of the user MAT (FMPAR) is set to general register R5. The start address of the program data storage area (FMPDR) is set to general register R4.

1. FMPAR setting

   FMPAR specifies the programming destination start address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the unit is 256 bytes, the lower eight bits (MOA7 to MOA0) must be in the 256-byte boundary of H'00.

2. FMPDR setting

   If the storage destination of the program data is flash memory, even when the program execution routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to on-chip RAM and then programming must be executed.

(2.11) Programming

There is an entry point of the programming program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programming is executed by using the following steps.

```
MOV.L  #DLTOP+16,R1          ; Set entry address to R1
JSR    @R1                   ; Call programming routine
NOP
```

1. The general registers other than R0 are saved in the programming program.
2. R0 is a return value of the FPFR parameter.
3. Since the stack area is used in the programming program, a stack area of maximum 128 bytes must be reserved in RAM.

(2.12) The return value in the programming program, FPFR (general register R0) is checked.

(2.13) Determine whether programming of the necessary data has finished.

If more than 256 bytes of data are to be programmed, specify FMPAR and FMPDR in 256-byte units, and repeat steps (2.10) to (2.13). Increment the programming destination address by 256 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(2.14) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of $\overline{\text{RES}}$ = 0) that is at least as long as the normal 100 μs.

(3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 21.12.

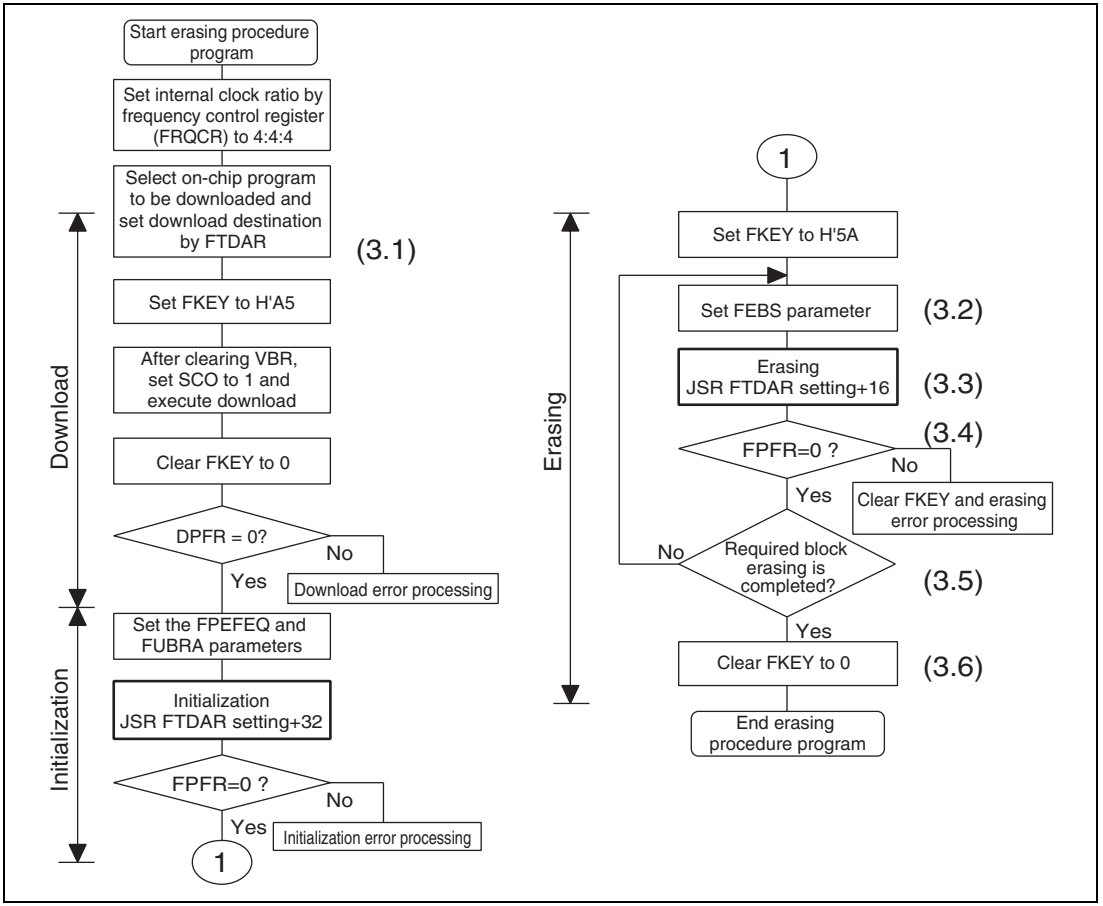


Figure 21.12 Erasing Procedure

The details of the erasing procedure are described below. The procedure program must be executed in an area other than the user MAT to be erased.

Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedural Program and Data for Programming.

The frequency division ratio of an internal clock (Iϕ), a bus clock (Bϕ), and a peripheral clock (Pϕ) is specified as 4:4:4 by the frequency control register (FRQCR).

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 21.10.

A single divided block is erased by one erasing processing. For block divisions, see figure 21.4. To erase two or more blocks, update the erase block number and perform the erasing processing for each block.

(3.1) Select the on-chip program to be downloaded and the download destination address

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see the description in section 21.5.2 (2), Programming Procedure in User Program Mode.

(3.2) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select parameter (FEBS: general register R4). If a value other than an erase block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFR.

(3.3) Erasure

Similar to as in programming, there is an entry point of the erasing program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and erasing is executed by using the following steps.

```
MOV.L  #DLTOP+16,R1         ; Set entry address to R1
JSR    @R1                  ; Call erasing routine
NOP
```

1. The general registers other than R0 are saved in the erasing program.
2. R0 is a return value of the FPFR parameter.
3. Since the stack area is used in the erasing program, a stack area of maximum 128 bytes must be reserved in RAM.

(3.4) The return value in the erasing program, FPFR (general register R0) is checked.

(3.5) Determine whether erasure of the necessary blocks has finished.

If more than one block is to be erased, update the FEBS parameter and repeat steps (3.2) to (3.5). Blocks that have already been erased can be erased again.

(3.6) After erasure finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT erasing has finished, secure a reset period (period of $\overline{RES}$ = 0) that is at least as long as the normal 100 µs.

(4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 21.13 shows an example of repetitively executing RAM emulation, erasing, and programming.

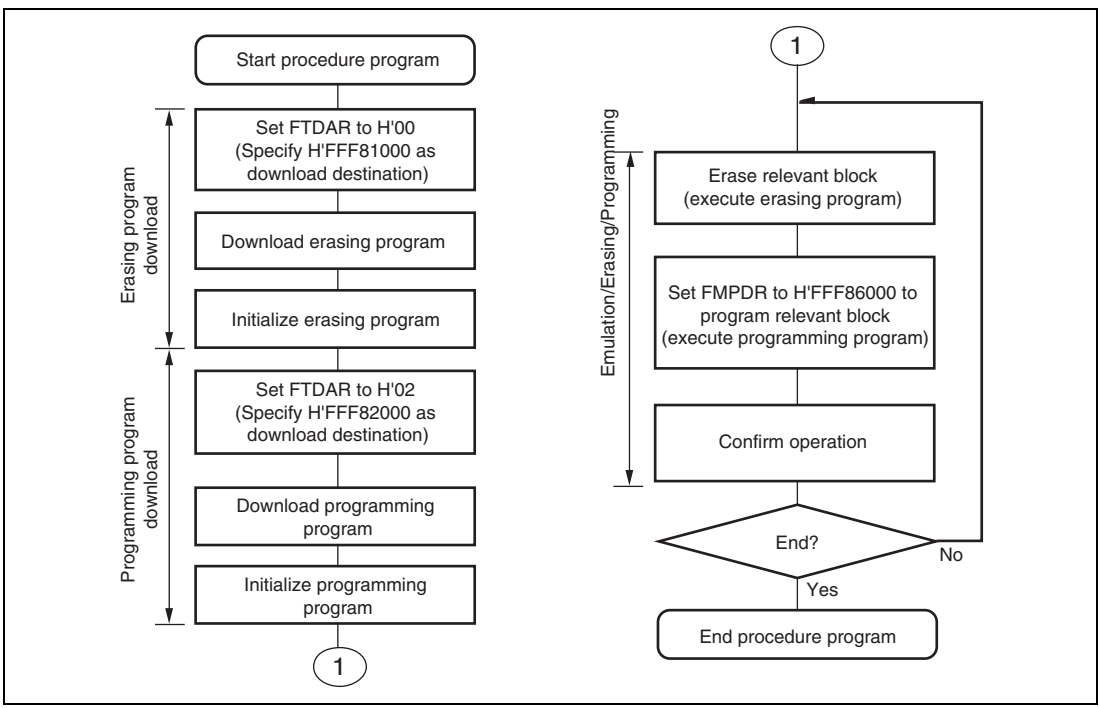


Figure 21.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)

Download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

1. Be careful not to destroy on-chip RAM with overlapped settings.

   In addition to the erasing program area and programming program area, areas for the user procedure programs, work area, and stack area are reserved in on-chip RAM. Do not make settings that will overwrite data in these areas.

2. Be sure to initialize both the erasing program and programming program.

   Initialization by setting the FPEFEQ and FUBRA parameters must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFF81020 in this example) and (download start address for programming program) + 32 bytes (H'FFF82020 in this example).

21.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than those in user program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCIF.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

(1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 21.1.

When the reset start is executed in user boot mode, the check routine for flash-memory related registers runs on the on-chip RAM. NMI and all other interrupts cannot be accepted. Neither can the AUD be used in this period. This period is 100 μs while operating at an internal frequency of 40 MHz.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to the flash MAT select register (FMATS) because the execution MAT is the user boot MAT.

(2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes. Figure 21.14 shows the procedure for programming the user MAT in user boot mode.

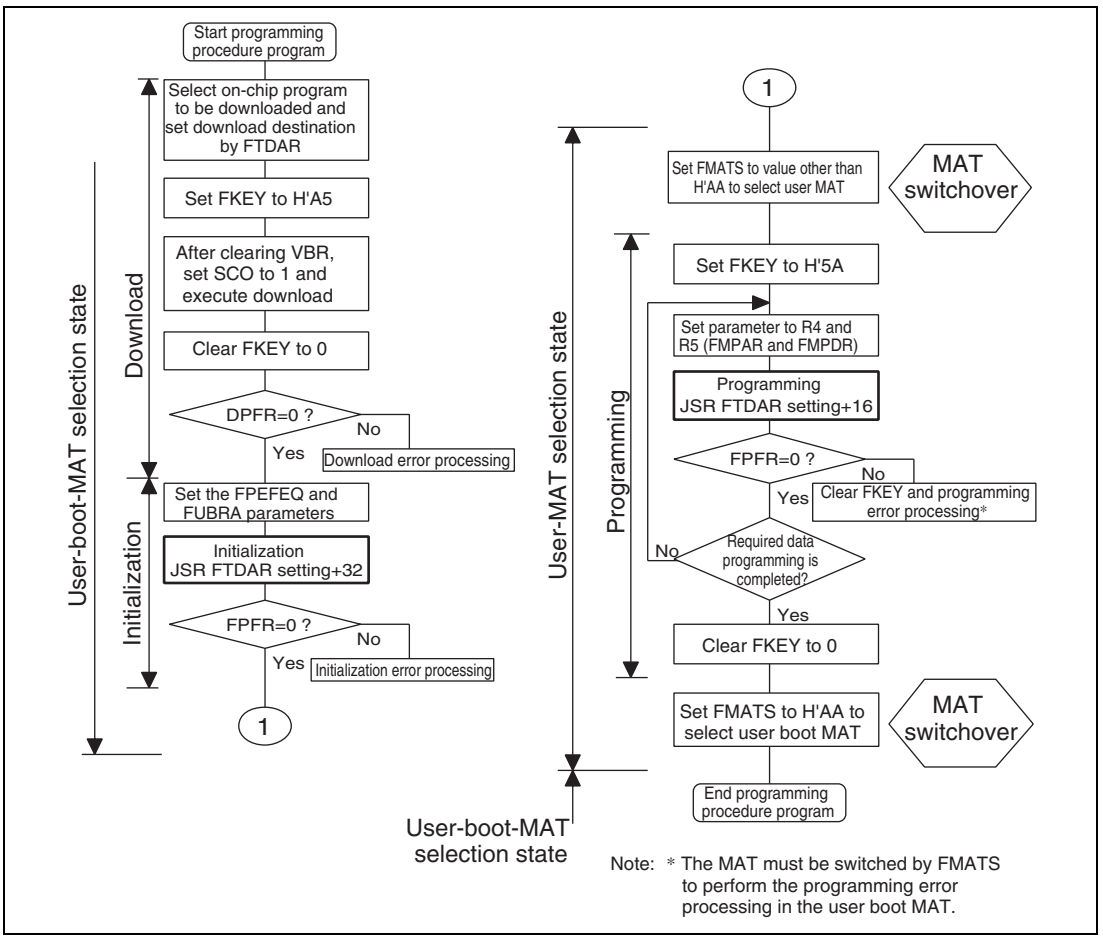


Figure 21.14 Procedure for Programming User MAT in User Boot Mode

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 21.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be located in an area other than flash memory. After programming finishes, switch the MATs again to return to the first state.

MAT switchover is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completely finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 21.7.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the programming procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedural Program and Data for Programming.

(3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 21.15 shows the procedure for erasing the user MAT in user boot mode.

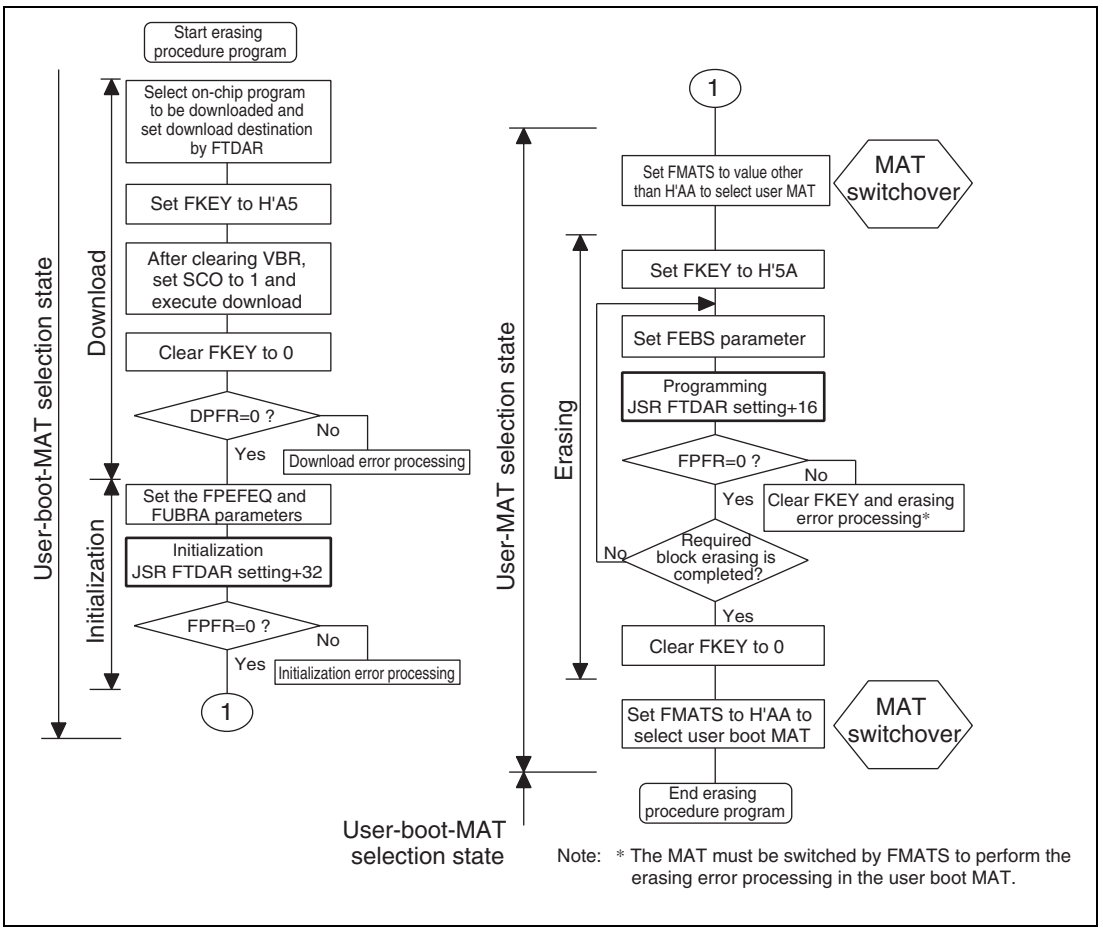


Figure 21.15 Procedure for Erasing User MAT in User Boot Mode

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 21.15.

MAT switching is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 21.7.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 21.8.2, Areas for Storage of the Procedural Program and Data for Programming.

21.6 Protection

There are three kinds of flash memory program/erase protection: hardware, software, and error protection.

21.6.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization of the flash memory are possible. However, an activated program for programming or erasure cannot program or erase locations in a user MAT, and the error in programming/erasing is reported in the FPFR parameter.

Table 21.8 Hardware Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasure**
FWE-pin protection	The input of a low-level signal on the FWE pin clears the FWE bit of FCCS and the LSI enters a programming/erasing-protected state.	—	√
Reset/standby protection	• A power-on reset (including a power-on reset by the WDT) and entry to standby mode initializes the programming/erasing interface registers and the LSI enters a programming/erasing-protected state. • Resetting by means of the $\overline{\text{RES}}$ pin after power is initially supplied will not make the LSI enter the reset state unless the $\overline{\text{RES}}$ pin is held low until oscillation has stabilized. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width that is specified in the section on AC characteristics. If the LSI is reset during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.	√	√

21.6.2 Software Protection

Software protection is set up in any of two ways: by disabling the downloading of on-chip programs for programming and erasing, and by means of a key code.

Table 21.9 Software Protection

		Function to be Protected	
Item	**Description**	**Download**	**Programming/ Erasure**
Protection by the SCO bit	Clearing the SCO bit in FCCS disables downloading of the programming/erasing program, thus making the LSI enter a programming/erasing-protected state.	√	√
Protection by FKEY	Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	√	√

21.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcontroller getting out of control during programming/erasing of the flash memory or operations that are not in accordance with the established procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcontroller malfunctions during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the LSI enters the error protection state, thus aborting programming or erasure.

The FLER bit is set to 1 in the following conditions:

- When the relevant bank area of flash memory is read during programming/erasing (including a vector read or an instruction fetch)
- When a SLEEP instruction (including software standby mode) is executed during programming/erasing

Error protection is cancelled (FLER bit is cleared) only by a power-on reset.

Note that the reset signal should only be released after providing a reset input over a period longer than the normal 100 µs. Since high voltages are applied during programming/erasing of the flash memory, some voltage may still remain even after the error protection state has been entered. For this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period so that the charge is released.

The state-transition diagram in figure 21.16 shows transitions to and from the error protection state.

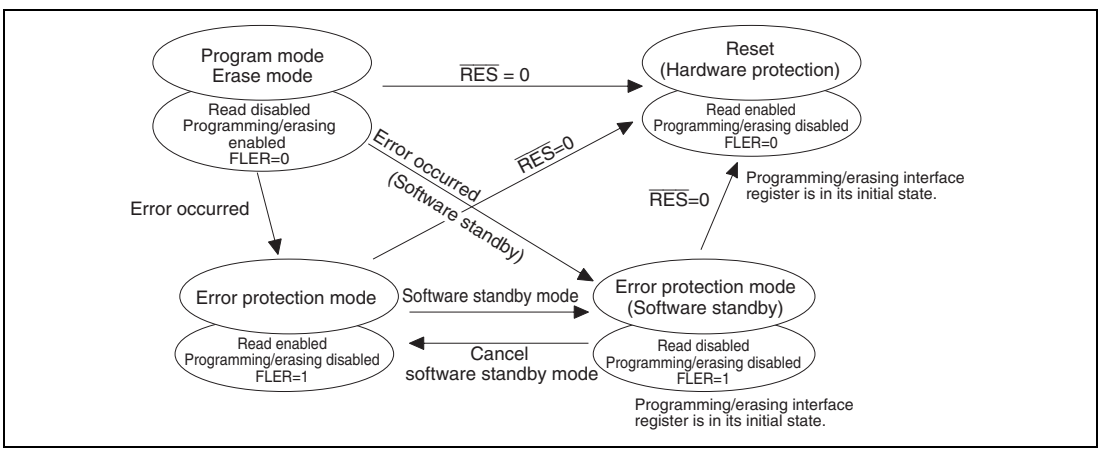


Figure 21.16 Transitions to and from Error Protection State

21.7 Usage Notes

21.7.1 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because these MATs are allocated to address 0.

(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT must take place in boot mode or programmer mode.)

1. MAT switching by FMATS should always be executed from the on-chip RAM. The SH microcontroller prefetches execution instructions. Therefore, a switchover during program execution in the user MAT causes an instruction code in the user MAT to be prefetched or an instruction in the newly selected user boot MAT to be prefetched, thus resulting in unstable operation.
2. To ensure that the MAT that has been switched to is accessible, execute thirty-two NOP instructions in on-chip RAM immediately after writing to FMATS of on-chip RAM (this prevents access to the flash memory during MAT switching).
3. If an interrupt occurs during switching, there is no guarantee of which memory MAT is being accessed.
   Always mask the maskable interrupts before switching MATs. In addition, configuring the system so that NMI interrupts do not occur during MAT switching is recommended.
4. After the MATs have been switched, take care because the interrupt vector table will also have been switched.
   If the same interrupt processings are to be executed before and after MAT switching or interrupt requests cannot be disabled, transfer the interrupt processing routine to on-chip RAM, and use the VBR setting to place the interrupt vector table in on chip RAM. In this case, make sure the VBR setting change does not conflict with the interrupt occurrence.
5. Memory sizes of the user MAT and user boot MAT are different. When accessing the user boot MAT, do not access addresses exceeding the 12-kbyte memory space. If access goes beyond the 12-kbyte space, the values read are undefined.

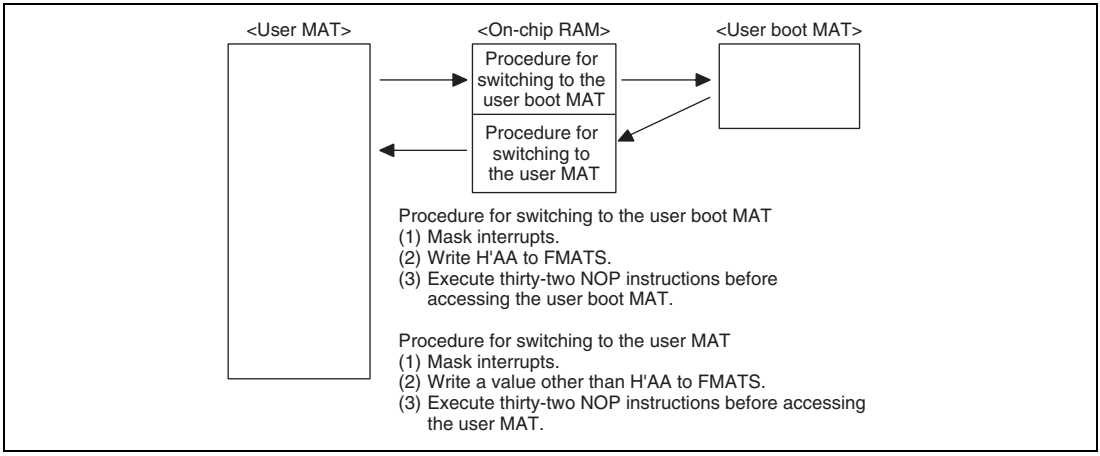


Figure 21.17 Switching between User MAT and User Boot MAT

21.7.2 Interrupts during Programming/Erasing

(1) Download of On-Chip Program

(a) VBR Setting Change

Before downloading the on-chip program, VBR must be set to H'80000000. If VBR is set to a value other than H'80000000, the interrupt vector table is placed in the user MAT (FMATS is not H'AA) or the user boot MAT (FMATS is H'AA) on setting H'80000000 to VBR.

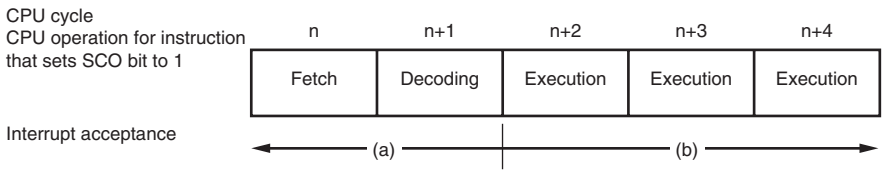
When VBR setting change conflicts with interrupt occurrence, whether the vector table before or after VBR is changed is referenced may cause an error.

Therefore, for cases where VBR setting change may conflict with interrupt occurrence, prepare a vector table to be referenced when VBR is H'00000000 (initial value) at the start of the user MAT or user boot MAT.

(b) SCO Download Request and Interrupt Request

Download of the on-chip programming/erasing program that is initiated by setting the SCO bit in FCCS to 1 generates a particular interrupt processing accompanied by MAT switchover. Operation when the SCO download request and interrupt request conflicts is described below.

1. Contention between SCO download request and interrupt request
   Figure 21.18 shows the timing of contention between execution of the instruction that sets the SCO bit in FCCS to 1 and interrupt acceptance.



(a) When the interrupt is accepted at the (n + 1) cycle or before
After the interrupt processing completes, the SCO bit is set to 1 and download is executed.

(b) When the interrupt is accepted at the (n + 2) cycle or later
The interrupt will conflicts with the SCO download request. Ensure that no interrupt is generated.

Figure 21.18 Timing of Contention between SCO Download Request and Interrupt Request

2. Generation of interrupt requests during downloading
Ensure that interrupts are not generated during downloading that is initiated by the SCO bit.

(2) Interrupts during Programming/Erasing

Though an interrupt processing can be executed at realtime during programming/erasing of the downloaded on-chip program, the following limitations and notes are applied.

1. When flash memory is being programmed or erased, both the user MAT and user boot MAT cannot be accessed. Prepare the interrupt vector table and interrupt processing routine in on-chip RAM or external memory. Make sure the flash memory being programmed or erased is not accessed by the interrupt processing routine. If flash memory is read, the read values are not guaranteed. If the relevant bank in flash memory that is being programmed or erased is accessed, the error protection state is entered, and programming or erasing is aborted. If a bank other than the relevant bank is accessed, the error protection state is not entered but the read values are not guaranteed.
2. Do not rewrite the program data specified by the FMPDR parameter. If new program data is to provided by the interrupt processing, temporarily save the new program data in another area. After confirming the completion of programming, save the new program data in the area specified by FMPDR or change the setting in FMPDR to indicated the other area in which the new program data was temporarily saved.
3. Make sure the interrupt processing routine does not rewrite the contents of the flash-memory related registers or data in the downloaded on-chip program area. During the interrupt processing, do not simultaneously perform download of the on-chip program by an SCO request or programming/erasing.
4. At the beginning of the interrupt processing routine, save the CPU register contents. Before returning from the interrupt processing, write the saved contents in the CPU registers again.
5. When a transition is made to sleep mode or software standby mode in the interrupt processing routine, the error protection state is entered and programming/erasing is aborted.
   If a transition is made to the reset state, the reset signal should only be released after providing a reset input over a period longer than the normal 100 μs to reduce the damage to flash memory.

21.7.3 Other Notes

(1) Download Time of On-Chip Program

The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 3 kbytes or less. Accordingly, when the CPU clock frequency is 40 MHz, the download for each program takes approximately 10 ms at maximum.

(2) User Branch Processing Intervals

The intervals for executing the user branch processing differs in programming and erasing. The processing phase also differs. Table 21.10 lists the maximum intervals for initiating the user branch processing when the CPU clock frequency is 40 MHz.

Table 21.10 Initiation Intervals of User Branch Processing

Processing Name	Maximum Interval
Programming	Approximately 2 ms*
Erasing	Approximately 15 ms*

Note: * Reference value

However, when operation is done with CPU clock of 40 MHz, maximum values of the time until first user branch processing are as shown in table 21.11.

Table 21.11 Initial User Branch Processing Time

Processing Name	Maximum
Programming	Approximately 2 ms*
Erasing	Approximately 15 ms*

Note: * Reference value

(3) Write to Flash-Memory Related Registers by DMAC

While an instruction in on-chip RAM is being executed, the DMAC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and destroy RAM or a MAT switchover may occur and the CPU get out of control.

(4) State in which Interrupts are Ignored

In the following modes or period, interrupt requests are ignored; they are not executed and the interrupt sources are not retained.

- Boot mode
- Programmer mode

(5) Compatibility with Programming/Erasing Program of Conventional F-ZTAT SH Microcontroller

A programming/erasing program for flash memory used in the conventional F-ZTAT SH microcontroller which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI.
Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.

(6) Monitoring Runaway by WDT

Unlike the conventional F-ZTAT SH microcontroller, no countermeasures are available for a runaway by WDT during programming/erasing by the downloaded on-chip program.
Prepare countermeasures (e.g. use of the user branch routine and periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.

(7) The Operating Frequency in On-Board Programming Mode

The frequency control register (FRQCR) should be set as follows in the erasing and programming procedure, described in section 21.5, On-Board Programming Mode.

- Specify (Iϕ: Bϕ: Pϕ) = (4: 4: 4) as the frequency division ratios of internal clocks.
  When the input clock is 10 MHz, (Iϕ: Bϕ: Pϕ) = (40 MHz: 40 MHz: 40MHz)
  When the input clock is 8 MHz, (Iϕ: Bϕ: Pϕ) = (32 MHz: 32 MHz: 32MHz)
- The following shows the frequency control register (FRQCR) values where (Iϕ: Bϕ: Pϕ) = (4: 4: 4) as the frequency division ratios of internal clocks.
  H'1000*
  H'1111*
  H'1333*

Note: * The CKOEN bit (bit 12) can be specified as either 0 or1.

(8) Programming the User MAT in User Program Mode

This LSI does not allow transitions from single chip mode to user program mode. Therefore, in order to program the user MAT in user program mode, be sure to activate the LSI in MCU extension mode 2 rather than in single chip mode.

21.8 Supplementary Information

21.8.1 Specifications of the Standard Serial Communications Interface in Boot Mode

The boot program activated in boot mode communicates with the host via the on-chip SCI of the LSI. The specifications of the serial communications interface between the host and the boot program are described below.

(1) States of Boot Program

The boot program has three states.

1. Bit-rate matching state

   In this state, the boot program adjusts the bit rate to match that of the host. When the chip starts up in boot mode, the boot program is activated and enters the bit-rate matching state, in which it receives commands from the host and adjusts the bit rate accordingly. After bit-rate matching is complete, the boot program proceeds to the inquiry-and-selection state.
2. Inquiry-and-selection state

   In this state, the boot program responds to inquiry commands from the host. The device, clock mode, and bit rate are selected in this state. After making these selections, the boot program enters the programming/erasure state in response to the transition-to-programming/erasure state command. The boot program transfers the erasure program to RAM and executes erasure of the user MAT and user boot MAT before it enters the programming/erasure state.
3. Programming/erasure state

   In this state, programming/erasure are executed. The boot program transfers the program for programming/erasure to RAM in line with the command received from the host and executes programming/erasure. It also performs sum checking and blank checking as directed by the respective commands.

Figure 21.19 shows the flow of processing by the boot program.

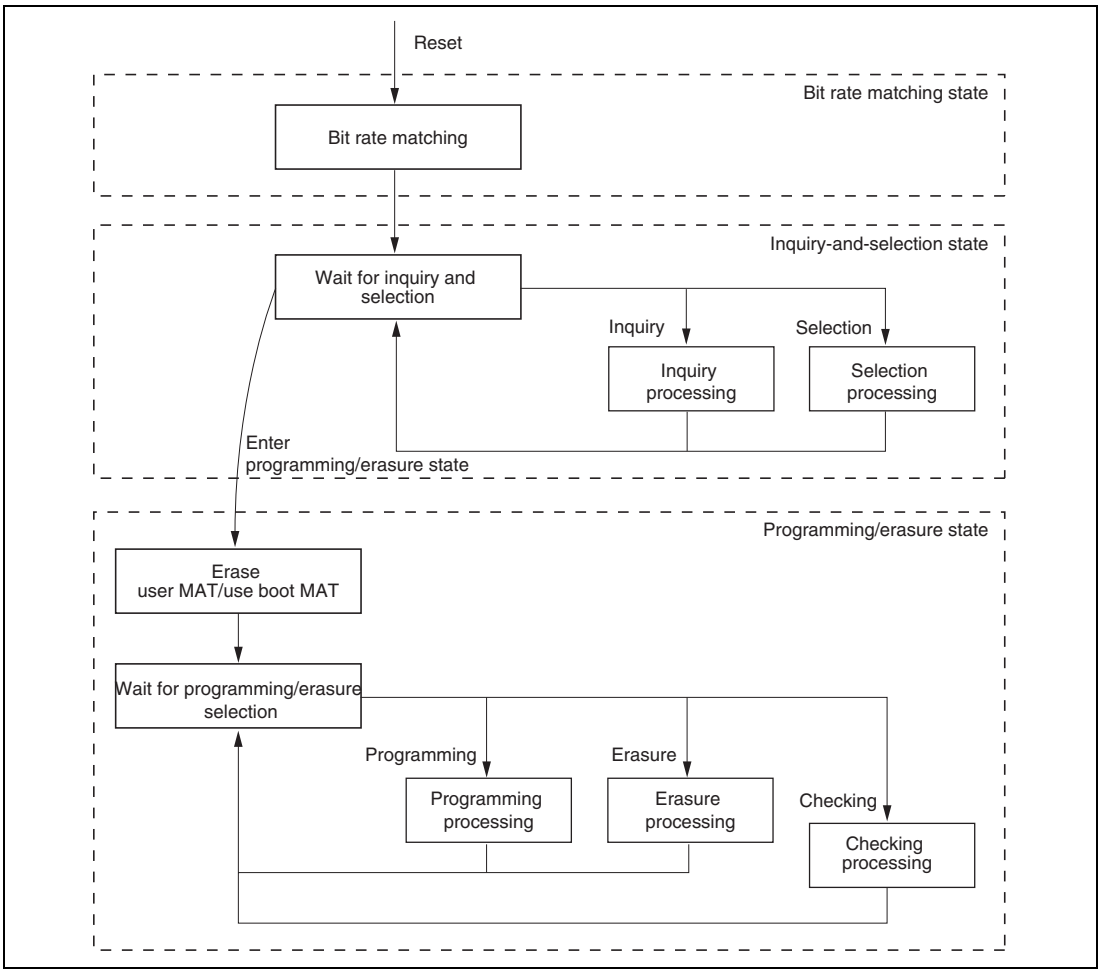


Figure 21.19 Flow of Processing by the Boot Program

(2) Bit-Rate Matching State

In bit-rate matching, the boot program measures the low-level intervals in a signal carrying H'00 data that is transmitted by the host, and calculates the bit rate from this. The bit rate can be changed by the new-bit-rate selection command. On completion of bit-rate matching, the boot program goes to the inquiry and selection state. The sequence of processing in bit-rate matching is shown in figure 21.20.

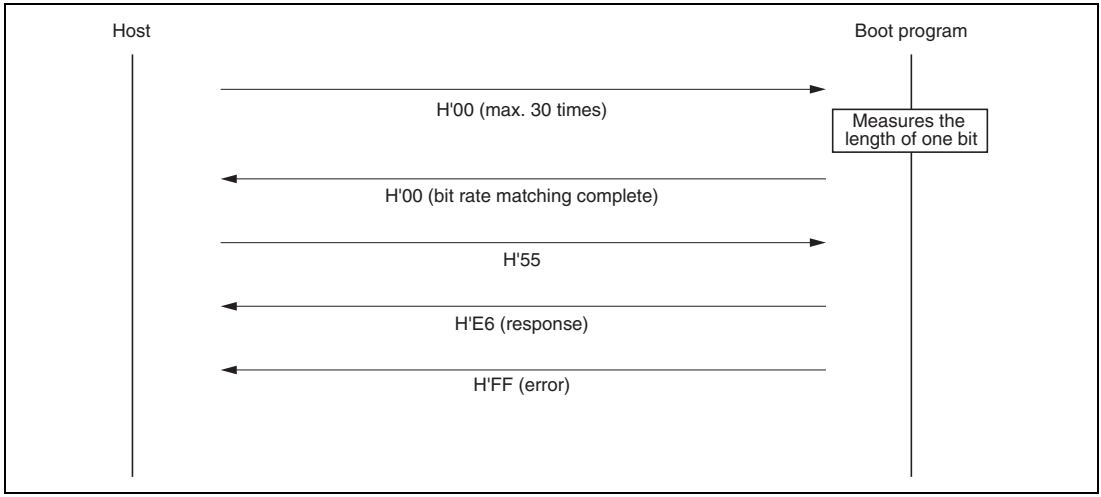


Figure 21.20 Sequence of Bit-Rate Matching

(3) Communications Protocol

Formats in the communications protocol between the host and boot program after completion of the bit-rate matching are as follows.

1. One-character command or one-character response

   A command or response consisting of a single character used for an inquiry or the ACK code indicating normal completion.
2. n-character command or n-character response

   A command or response that requires n bytes of data, which is used as a selection command or response to an inquiry. The length of programming data is treated separately below.
3. Error response

   Response to a command in case of an error: two bytes, consisting of the error response and error code.
4. 256-byte programming command

   The command itself does not include data-size information. The data length is known from the response to the command for inquiring about the programming size.
5. Response to a memory reading command

   This response includes four bytes of size information.

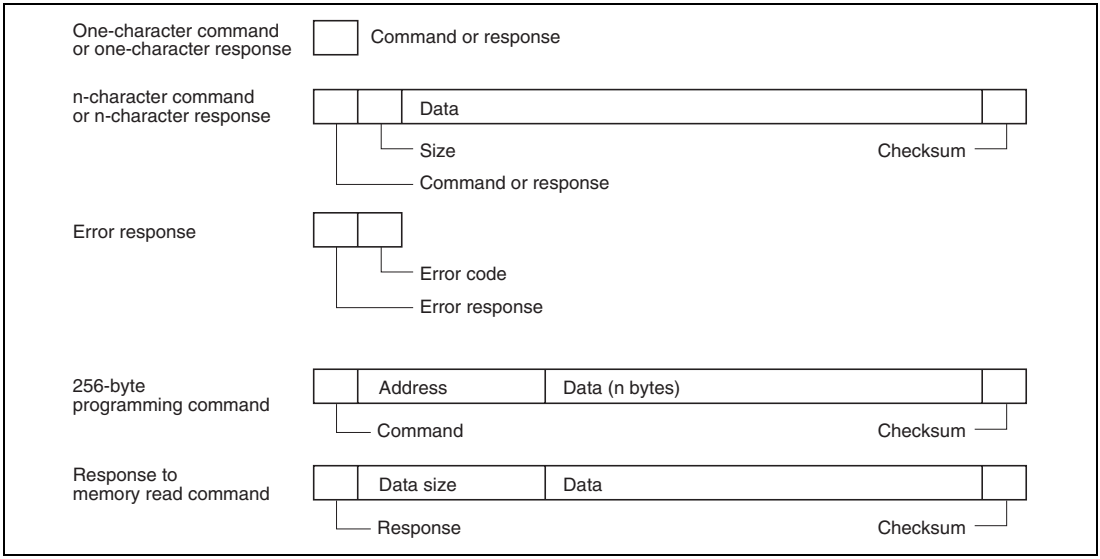


Figure 21.21 Formats in the Communications Protocol

- Command (1 byte): Inquiry, selection, programming, erasure, checking, etc.
- Response (1 byte): Response to an inquiry
- Size (one or two bytes): The length of data for transfer, excluding the command/response, size, and checksum.
- Data (n bytes): Particular data for the command or response
- Checksum (1 byte): Set so that the total sum of byte values from the command code to the checksum is H'00.
- Error response (1 byte): Error response to a command
- Error code (1 byte): Indicates the type of error.
- Address (4 bytes): Address for programming
- Data (n bytes): Data to be programmed. "n" is known from the response to the command used to inquire about the programming size.
- Data size (4 bytes): Four-byte field included in the response to a memory reading command.

(4) Inquiry-and-Selection State

In this state, the boot program returns information on the flash ROM in response to inquiry commands sent from the host, and selects the device, clock mode, and bit rate in response to the respective selection commands.

The inquiry and selection commands are listed in table 21.12.

Table 21.12 Inquiry and Selection Commands

Command	Command Name	Function
H'20	Inquiry on supported devices	Requests the device codes and their respective boot program names.
H'10	Device selection	Selects a device code.
H'21	Inquiry on clock modes	Requests the number of available clock modes and their respective values.
H'11	Clock-mode selection	Selects a clock mode.
H'22	Inquiry on frequency multipliers	Requests the number of clock signals for which frequency multipliers and divisors are selectable, the number of multiplier and divisor settings for the respective clocks, and the values of the multipliers and divisors.
H'23	Inquiry on operating frequency	Requests the minimum and maximum values for operating frequency of the main clock and peripheral clock.
H'24	Inquiry on user boot MATs	Requests the number of user boot MAT areas along with their start and end addresses.
H'25	Inquiry on user MATs	Requests the number of user MAT areas along with their start and end addresses.
H'26	Inquiry on erasure blocks	Requests the number of erasure blocks along with their start and end addresses.
H'27	Inquiry on programming size	Requests the unit of data for programming.
H'3F	New bit rate selection	Selects a new bit rate.
H'40	Transition to programming/erasure state	On receiving this command, the boot program erases the user MAT and user boot MAT and enters the programming/erasure state.
H'4F	Inquiry on boot program state	Requests information on the current state of boot processing.

The selection commands should be sent by the host in this order: device selection (H'10), clock-mode selection (H'11), new bit rate selection (H'3F). These commands are mandatory. If the same selection command is sent two or more times, the command that is sent last is effective.

All commands in the above table, except for the boot program state inquiry command (H'4F), are valid until the boot program accepts the transition-to-programming/erasure state command (H'40). That is, until the transition command is accepted, the host can continue to send commands listed in the above table until it has made the necessary inquiries and selections. The host can send the boot program state inquiry command (H'4F) even after acceptance of the transition-to-programming/erasure state command (H'40) by the boot program.

(a) Inquiry on Supported Devices

In response to the inquiry on supported devices, the boot program returns the device codes of the devices it supports and the product names of their respective boot programs.

Command	H'20

- Command H'20 (1 byte): Inquiry on supported devices

Response	H'30	Size	No. of devices	
	Number of characters	Device code		Product name
	…			
	SUM			

- Response H'30 (1 byte): Response to the inquiry on supported devices
- Size (1 byte): The length of data for transfer excluding the command code, this field (size), and the checksum. Here, it is the total number of bytes taken up by the number of devices, number of characters, device code, and product name fields.
- Number of devices (1 byte): The number of device models supported by the boot program embedded in the microcontroller.
- Number of characters (1 byte): The number of characters in the device code and product name fields.
- Device code (4 bytes): Device code of a supported device (ASCII encoded)
- Product name (n bytes): Product code of the boot program (ASCII encoded)
- SUM (1 byte): Checksum
  This is set so that the total sum of all bytes from the command code to the checksum is H'00.

(b) Device Selection

In response to the device selection command, the boot program sets the specified device as the selected device. The boot program will return the information on the selected device in response to subsequent inquiries.

Command	H'10	Size	Device code	SUM

- Command H'10 (1 byte): Device selection
- Size (1 byte): Number of characters in the device code (fixed at 4)
- Device code (4 bytes): A device code that was returned in response to an inquiry on supported devices (ASCII encoded)
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to device selection
  This is the ACK code and is returned when the specified device code matches one of the supported devices.

Error response	H'90	ERROR

- Error response H'90 (1 byte): Error response to device selection
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'21: Non-matching device code

(c) Inquiry on Clock Modes

In response to the inquiry on clock modes, the boot program returns the number of available clock modes.

Command	H'21

- Command H'21 (1 byte): Inquiry on clock modes

Response	H'31	Size	Mode	…	SUM

- Response H'31 (1 byte): Response to the inquiry on clock modes
- Size (1 byte): The total length of the number of modes and mode data fields. (fixed at 1)
- Mode (1 byte): Selectable clock mode
- SUM (1 byte): Checksum

(d) Clock-Mode Selection

In response to the clock-mode selection command, the boot program sets the specified clock mode. The boot program will return the information on the selected clock mode in response to subsequent inquiries.

Command	H'11	Size	Mode	SUM

- Command H'11 (1 byte): Clock mode selection
- Size (1 byte): Number of characters in the clock-mode field (fixed at 1)
- Mode (1 byte): A clock mode returned in response to the inquiry on clock modes
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to clock mode selection
  This is the ACK code and is returned when the specified clock-mode matches one of the available clock modes.

Error response	H'91	ERROR

- Error response H'91 (1 byte): Error response to clock mode selection
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'21: Non-matching clock mode

(e) Inquiry on Frequency Multipliers

In response to the inquiry on frequency multipliers, the boot program returns information on the settable frequency multipliers or divisors.

Command	H'22

- Command H'22 (1 byte): Inquiry on frequency multipliers

Response	H'32	Size	Number of operating clocks					
	No. of multipliers	Multiplier	…					
	…							
	SUM							

- Response H'32 (1 byte): Response to the inquiry on frequency multipliers
- Size (1 byte): The total length of the number of operating clocks, number of multipliers, and multiplier fields.
- Number of operating clocks (1 byte): The number of operating clocks for which multipliers can be selected
  (for example, if frequency multiplier settings can be made for the frequencies of the main and peripheral operating clocks, the value should be H'02).
- Number of multipliers (1 byte): The number of multipliers selectable for the operating frequency of the main or peripheral modules
- Multiplier (1 byte):
  Multiplier: Numerical value in the case of frequency multiplication (e.g. H'04 for ×4)
  Divisor: Two's complement negative numerical value in the case of frequency division (e.g. H'FE [-2] for ×1/2)
  As many multiplier fields are included as there are multipliers or divisors, and combinations of the number of multipliers and multiplier fields are repeated as many times as there are operating clocks.
- SUM (1 byte): Checksum

(f) Inquiry on Operating Frequency

In response to the inquiry on operating frequency, the boot program returns the number of operating frequencies and the maximum and minimum values.

Command	H'23

- Command H'23 (1 byte): Inquiry on operating frequency

Response	H'33	Size	Number of operating clocks
	Operating freq. (min)		Operating freq. (max)
	…		
	SUM		

- Response H'33 (1 byte): Response to the inquiry on operating frequency
- Size (1 byte): The total length of the number of operating clocks, and maximum and minimum values of operating frequency fields.
- Number of operating clocks (1 byte): The number of operating clock frequencies required within the device.
  For example, the value two indicates main and peripheral operating clock frequencies.
- Minimum value of operating frequency (2 bytes): The minimum frequency of a frequency-multiplied or -divided clock signal.
  The value in this field and in the maximum value field is the frequency in MHz to two decimal places, multiplied by 100 (for example, if the frequency is 20.00 MHz, the value multiplied by 100 is 2000, so H'07D0 is returned here).
- Maximum value of operating frequency (2 bytes): The maximum frequency of a frequency-multiplied or -divided clock signal.
  As many pairs of minimum/maximum values are included as there are operating clocks.
- SUM (1 byte): Checksum

(g) Inquiry on User Boot MATs

In response to the inquiry on user boot MATs, the boot program returns the number of user boot MAT areas and their addresses.

Command	H'24

- Command H'24 (1 byte): Inquiry on user boot MAT information

Response	H'34	Size	No. of areas	
	First address of the area			Last address of the area
	…			
	SUM			

- Response H'34 (1 byte): Response to the inquiry on user boot MATs
- Size (1 byte): The total length of the number of areas and first and last address fields.
- Number of areas (1 byte): The number of user boot MAT areas.
  H'01 is returned if the entire user boot MAT area is continuous.
- First address of the area (4 bytes)
- Last address of the area (4 bytes)
  As many pairs of first and last address field are included as there are areas.
- SUM (1 byte): Checksum

(h) Inquiry on User MATs

In response to the inquiry on user MATs, the boot program returns the number of user MAT areas and their addresses.

Command	H'25

- Command H'25 (1 byte): Inquiry on user MAT information

Response	H'35	Size	No. of areas	
	First address of the area			Last address of the area
	…			
	SUM			

- Response H'35 (1 byte): Response to the inquiry on user MATs
- Size (1 byte): The total length of the number of areas and first and last address fields.
- Number of areas (1 byte): The number of user MAT areas.
  H'01 is returned if the entire user MAT area is continuous.
- First address of the area (4 bytes)
- Last address of the area (4 bytes)
  As many pairs of first and last address field are included as there are areas.
- SUM (1 byte): Checksum

(i) Inquiry on Erasure Blocks

In response to the inquiry on erasure blocks, the boot program returns the number of erasure blocks in the user MAT and the addresses where each block starts and ends.

Command	H'26

- Command H'26 (1 byte): Inquiry on erasure blocks

Response	H'36	Size	No. of blocks	
	First address of the block			Last address of the block
	…			
	SUM			

- Response H'36 (1 byte): Response to the inquiry on erasure blocks
- Size (2 bytes): The total length of the number of blocks and first and last address fields.
- Number of blocks (1 byte): The number of erasure blocks in flash memory
- First address of the block (4 bytes)
- Last address of the block (4 bytes)
  As many pairs of first and last address data are included as there are blocks.
- SUM (1 byte): Checksum

(j) Inquiry on Programming Size

In response to the inquiry on programming size, the boot program returns the size, in bytes, of the unit for programming.

Command	H'27	

- Command H'27 (1 byte): Inquiry on programming size

Response	H'37	Size	Programming size	SUM

- Response H'37 (1 byte): Response to the inquiry on programming size
- Size (1 byte): The number of characters in the programming size field (fixed at 2)
- Programming size (2 bytes): The size of the unit for programming
  This is the unit for the reception of data to be programmed.
- SUM (1 byte): Checksum

(k) New Bit Rate Selection

In response to the new-bit-rate selection command, the boot program changes the bit rate setting to the new bit rate and, if the setting was successful, responds to the ACK sent by the host by returning another ACK at the new bit rate.

The new-bit-rate selection command should be sent after clock-mode selection.

Command	H'3F	Size	Bit rate	Input frequency
	No. of multipliers	Multiplier 1	Multiplier 2	
	SUM			

- Command H'3F (1 byte): New bit rate selection
- Size (1 byte): The total length of the bit rate, input frequency, number of multipliers, and multiplier fields
- Bit rate (2 bytes): New bit rate
  The bit rate value divided by 100 should be set here (for example, to select 19200 bps, the set H'00C0, which is 192 in decimal notation).
- Input frequency (2 bytes): The frequency of the clock signal fed to the boot program
  This should be the frequency in MHz to the second decimal place, multiplied by 100 (for example, if the frequency is 8.882 MHz, the value is truncated to the second decimal place and multiplied by 100, making 888; so H'0378 should be set in this field).

- Number of multipliers (1 byte): The number of selectable frequency multipliers and divisors for the device.
  This is normally 2, which indicates the main operating frequency and the operating frequency of the peripheral modules.
- Multiplier 1 (1 byte): Multiplier or divisor for the main operating frequency
  Multiplier: Numerical value of the frequency multiplier (e.g. H'04 for ×4)
  Divisor: Two's complement negative numerical value in the case of frequency division (e.g. H'FE [-2] for ×1/2)
- Multiplier 2 (1 byte): Multiplier or divisor for the peripheral operating frequency
  Multiplier: Numerical value of the frequency multiplier (e.g. H'04 for ×4)
  Divisor: Two's complement negative numerical value in the case of frequency division (e.g. H'FE [-2] for ×1/2)
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to the new-bit-rate selection command
  This is the ACK code and is returned if the specified bit rate is selected.

Error response	H'BF	ERROR

- Error response H'BF (1 byte): Error response to the new-bit-rate selection command
- ERROR (1 byte): Error code

  H'11: Sum-check error

  H'24: Bit rate selection error (the specified bit rate is not selectable).

  H'25: Input frequency error (the specified input frequency is not within the range from the minimum to the maximum value).

  H'26: Frequency multiplier error (the specified multiplier does not match an available one).

  H'27: Operating frequency error (the specified operating frequency is not within the range from the minimum to the maximum value).

The received data are checked in the following ways.

1. Input frequency

   The value of the received input frequency is checked to see if it is within the range of the minimum and maximum values of input frequency for the selected clock mode of the selected device. A value outside the range generates an input frequency error.

2. Multiplier

   The value of the received multiplier is checked to see if it matches a multiplier or divisor that is available for the selected clock mode of the selected device. A value that does not match an available ratio generates a frequency multiplier error.

3. Operating frequency

   The operating frequency is calculated from the received input frequency and the frequency multiplier or divisor. The input frequency is the frequency of the clock signal supplied to the LSI, while the operating frequency is the frequency at which the LSI is actually driven. The following formulae are used for this calculation.

   Operating frequency = input frequency × multiplier, or

   Operating frequency = input frequency / divisor

   The calculated operating frequency is checked to see if it is within the range of the minimum and maximum values of the operating frequency for the selected clock mode of the selected device. A value outside the range generates an operating frequency error.

4. Bit rate

   From the peripheral operating frequency (Pϕ) and the bit rate (B), the value (= n) of the clock select bits (CKS) in the serial mode register (SCSMR) and the value (= N) of the bit rate register (SCBRR) are calculated, after which the error in the bit rate is calculated. This error is checked to see if it is smaller than 4%. A result greater than or equal to 4% generates a bit rate selection error. The following formula is use to calculate the error.

$$\text{Error (\%)} = \left\{ \left[\frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} \right] - 1 \right\} \times 100$$

When the new bit rate is selectable, the boot program returns an ACK code to the host and then makes the register setting to select the new bit rate. The host then sends an ACK code at the new bit rate, and the boot program responds to this with another ACK code, this time at the new bit rate.

Acknowledge	H'06

- Acknowledge H'06 (1 byte): The ACK code sent by the host to acknowledge the new bit rate.

Response	H'06

- Response H'06 (1 byte): The ACK code transferred in response to acknowledgement of the new bit rate

The sequence of new bit rate selection is shown in figure 21.22.

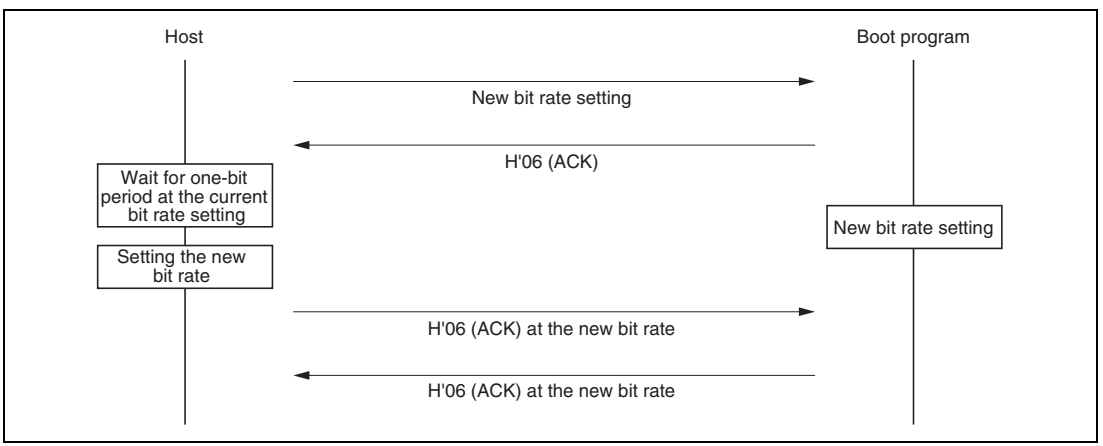


Figure 21.22 Sequence of New Bit Rate Selection

(l) Transition to the Programming/Erasure State

In response to the transition to the programming/erasure state command, the boot program transfers the erasing program and runs it to erase any data in the user MAT and then the user boot MAT. On completion of this erasure, the boot program returns the ACK code and enters the programming/erasure state.

Before sending the programming selection command and data for programming, the host must select the device, clock mode, and new bit rate for the LSI by issuing the device selection command, clock-mode selection command, new-bit-rate selection command, and then initiate the transition to the programming/erasure state by sending the corresponding command to the boot program.

Command	H'40

- Command H'40 (1 byte): Transition to programming/erasure state

Response	H'06

- Response H'06 (1 byte): Response to the transition-to-programming/erasure state command
  This is returned as ACK when erasure of the user boot MAT and user MAT has succeeded after transfer of the erasure program.

Error response	H'C0	H'51

- Error response H'C0 (1 byte): Error response to the transition-to-programming/erasure state command
- ERROR (1 byte): Error code
  H'51: Erasure error (Erasure did not succeed because of an error.)

(5) Command Error

Command errors are generated by undefined commands, commands sent in an incorrect order, and the inability to accept a command. For example, sending the clock-mode selection command before device selection or an inquiry command after the transition-to-programming/erasure state command generates a command error.

Error response	H'80	H'xx

- Error response H'80 (1 byte): Command error
- Command H'xx (1 byte): Received command

(6) Order of Commands

In the inquiry-and-selection state, commands should be sent in the following order.

1. Send the inquiry on supported devices command (H'20) to get the list of supported devices.
2. Select a device from the returned device information, and send the device selection command (H'10) to select that device.
3. Send the inquiry on clock mode command (H'21) to get the available clock modes.
4. Select a clock mode from among the returned clock modes, and send the clock-mode selection command (H'11).
5. After selection of the device and clock mode, send the commands to inquire about frequency multipliers (H'22) and operating frequencies (H'23) to get the information required to select a new bit rate.
6. Taking into account the returned information on the frequency multipliers and operating frequencies, send a new-bit-rate selection command (H'3F).
7. After the device and clock mode have been selected, get the information required for programming and erasure of the user boot MAT and user MAT by sending the commands to inquire about the user boot MAT (H'24), user MAT (H'25), erasure block (H'26), and programming size (H'27).
8. After making all necessary inquiries and the new bit rate selection, send the transition-to-programming/erasure state command (H'40) to place the boot program in the programming/erasure state.

(7) Programming/Erasure State

In this state, the boot program must select the form of programming corresponding to the programming-selection command and then write data in response to 256-byte programming commands, or perform erasure in block units in response to the erasure-selection and block-erasure commands.

The programming and erasure commands are listed in table 21.13.

Table 21.13 Programming and Erasure Commands

Command	Command Name	Function
H'42	Selection of user boot MAT programming	Selects transfer of the program for user boot MAT programming.
H'43	Selection of user MAT programming	Selects transfer of the program for user MAT programming.
H'50	256-byte programming	Executes 256-byte programming.
H'48	Erasure selection	Selects transfer of the erasure program.
H'58	Block erasure	Executes erasure of the specified block.
H'52	Memory read	Reads from memory.
H'4A	Sum checking of user boot MAT	Executes sum checking of the user boot MAT.
H'4B	Sum checking of user MAT	Executes sum checking of the user MAT.
H'4C	Blank checking of user boot MAT	Executes blank checking of the user boot MAT.
H'4D	Blank checking of user MAT	Executes blank checking of the user MAT.
H'4F	Inquiry on boot program state	Requests information on the state of boot processing.

(8) Programming

Programming is performed by issuing a programming-selection command and the 256-byte programming command.

Firstly, the host issues the programming-selection command to select the MAT to be programmed. Two programming-selection commands are provided for the selection of either of the two target areas.

1. Selection of user boot MAT programming
2. Selection of user MAT programming

Next, the host issues a 256-byte programming command. 256 bytes of data for programming by the method selected by the preceding programming selection command are expected to follow the command. To program more than 256 bytes, repeatedly issue 256-byte programming commands. To terminate programming, the host should send another 256-byte programming command with the address H'FFFFFFFF. On completion of programming, the boot program waits for the next programming/erasure selection command.

To then program the other MAT, start by sending the programming select command.

The sequence of programming by programming-selection and 256-byte programming commands is shown in figure 21.23.

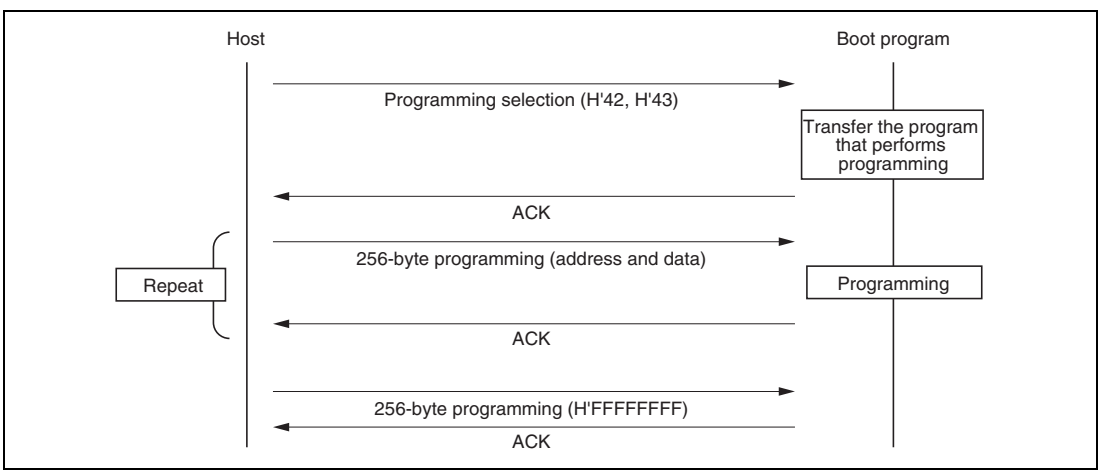


Figure 21.23 Sequence of Programming

(a) Selection of User Boot MAT Programming

In response to the command for selecting programming of the user boot MAT, the boot program transfers the corresponding flash-writing program, i.e. the program for writing to the user boot MAT.

Command	H'42

- Command H'42 (1 byte): Selects programming of the user boot MAT.

Response	H'06

- Response H'06 (1 byte): Response to selection of user boot MAT programming
  This ACK code is returned after transfer of the program that performs writing to the user boot MAT.

Error response	H'C2	ERROR

- Error response H'C2 (1 byte): Error response to selection of user boot MAT programming
- ERROR (1 byte): Error code
  H'54: Error in selection processing (processing was not completed because of a transfer error)

(b) Selection of User MAT Programming

In response to the command for selecting programming of the user MAT, the boot program transfers the corresponding flash-writing program, i.e. the program for writing to the user MAT.

Command	H'43

- Command H'43 (1 byte): Selects programming of the user MAT.

Response	H'06

- Response H'06 (1 byte): Response to selection of user MAT programming
  This ACK code is returned after transfer of the program that performs writing to the user MAT.

Error response		
	H'C3	ERROR

- Error response H'C3 (1 byte): Error response to selection of user MAT programming
- ERROR (1 byte): Error code
  H'54: Error in selection processing (processing was not completed because of a transfer error)

(c) 256-Byte Programming

In response to the 256-byte programming command, the boot program executes the flash-writing program transferred in response to the command to select programming of the user boot MAT or user MAT.

Command								
	H'50	Address for programming						
	Data	…						
	…							
	SUM							

- Command H'50 (1 byte): 256-byte programming
- Address for programming (4 bytes): Address where programming starts
  Specify the address of a 256-byte boundary.
  [Example] H'00, H01, H'00, H'00: H'00010000
- Programming data (n bytes): Data for programming
  The length of the programming data is the size returned in response to the programming size inquiry command.
- SUM (1 byte): Checksum

Response	
	H'06

- Response H'06 (1 byte): Response to 256-byte programming
  The ACK code is returned on completion of the requested programming.

Error response		
	H'D0	ERROR

- Error response H'D0 (1 byte): Error response to 256-byte programming
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'2A: Address error (the address is not within the range for the selected MAT)
  H'53: Programming error (programming failed because of an error in programming)

Specify H'00 for the lower byte of the address on a boundary corresponding to the unit of programming (programming size). When less than 256 bytes of data are to be programmed, the host should transmit the data after padding the vacant bytes with H'FF.

To terminate programming of a given MAT, send a 256-byte programming command with the address field H'FFFFFFFF. This informs the boot program that all data for the selected MAT have been sent; the boot program then waits for the next programming/erasure selection command.

Command	H'50	Address for programming	SUM

- Command H'50 (1 byte): 256-byte programming
- Address for programming (4 bytes): Terminating code (H'FF, H'FF, H'FF, H'FF)
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to 256-byte programming
  This ACK code is returned on completion of the requested programming.

Error response	H'D0	ERROR

- Error response H'D0 (1 byte): Error response to 256-byte programming
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'53: Programming error

(9) Erasure

Erasure is performed by issuing the erasure selection command and then one or more block erasure commands.

Firstly, the host sends the erasure selection command to select erasure; after that, it sends a block erasure command to actually erase a specific block. To erase multiple blocks, send further block erasure commands. To terminate erasure, the host should send a block erasure command with the block number H'FF. After this, the boot program waits for the next programming/erasure selection command.

The sequence of erasure by the erasure selection command and block erasure command is shown in figure 21.24.

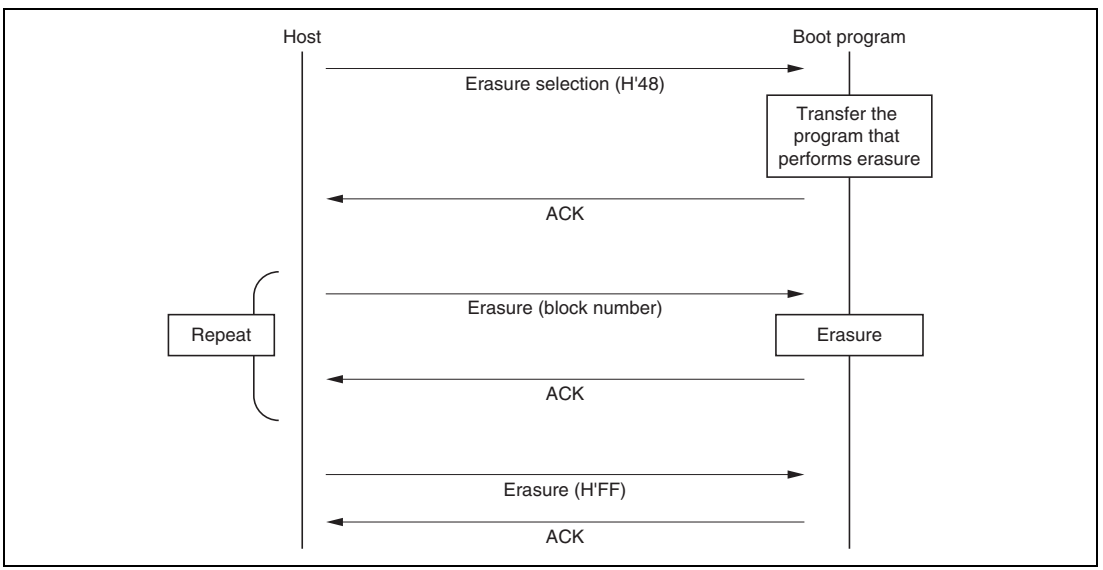


Figure 21.24 Sequence of Erasure

(a) Select Erasure

In response to the erasure selection command, the boot program transfers the program that performs erasure, i.e. erases data in the user MAT.

Command	H'48

- Command H'48 (1 byte): Selects erasure.

Response	H'06

- Response H'06 (1 byte): Response to selection of erasure
  This ACK code is returned after transfer of the program that performs erasure.

Error response	H'C8	ERROR

- Error response H'C8 (1 byte): Error response to selection of erasure
- ERROR (1 byte): Error code
  H'54: Error in selection processing (processing was not completed because of a transfer error.)

(b) Block Erasure

In response to the block erasure command, the boot program erases the data in a specified block of the user MAT.

Command	H'58	Size	Block number	SUM

- Command H'58 (1 byte): Erasure of a block
- Size (1 byte): The number of characters in the block number field (fixed at 1)
- Block number (1 byte): Block number of the block to be erased
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): Response to the block erasure command
  This ACK code is returned when the block has been erased.

Error response	H'D8	ERROR

- Error response H'D8 (1 byte): Error response to the block erasure command
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'29: Block number error (the specified block number is incorrect.)
  H'51: Erasure error (an error occurred during erasure.)

On receiving the command with H'FF as the block number, the boot program stops erasure processing and waits for the next programming/erasure selection command.

Command	H'58	Size	Block number	SUM

- Command H'58 (1 byte): Erasure of a block
- Size (1 byte): The number of characters in the block number field (fixed at 1)
- Block number (1 byte): H'FF (erasure terminating code)
- SUM (1 byte): Checksum

Response	H'06

- Response H'06 (1 byte): ACK code to indicate response to the request for termination of erasure

To perform erasure again after having issued the command with the block number specified as H'FF, execute the process from the selection of erasure.

(10) Memory Read

In response to the memory read command, the boot program returns the data from the specified address.

Command	H'52	Size	Area	First address for reading	
	Amount to read				SUM

- Command H'52 (1 byte): Memory read
- Size (1 byte): The total length of the area, address for reading, and amount to read fields (fixed value of 9)
- Area (1 byte):
  H'00: User boot MAT
  H'01: User MAT
  An incorrect area specification will produce an address error.
- Address where reading starts (4 bytes)
- Amount to read (4 bytes): The amount of data to be read
- SUM (1 byte): Checksum

Response	H'52	Amount to read						
	Data	…						
	SUM							

- Response H'52 (1 byte): Response to the memory read command
- Amount to read (4 bytes): The amount to read as specified in the memory read command
- Data (n bytes): The specified amount of data read out from the specified address
- SUM (1 byte): Checksum

Error response	H'D2	ERROR

- Error response H'D2 (1 byte): Error response to memory read command
- ERROR (1 byte): Error code
  H'11: Sum-check error
  H'2A: Address error (the address specified for reading is beyond the range of the MAT)
  H'2B: Size error (the specified amount is greater than the size of the MAT,
  the last address for reading as calculated from the specified address for the start of reading and the amount to read is beyond the MAT area, or "0" was specified as the amount to read)

(11) Sum Checking of the User Boot MAT

In response to the command for sum checking of the user boot MAT, the boot program adds all bytes of data in the user boot MAT and returns the result.

Command	H'4A

- Command H'4A (1 byte): Sum checking of the user boot MAT

Response	H'5A	Size	Checksum for the MAT	SUM

- Response H'5A (1 byte): Response to sum checking of the user boot MAT
- Size (1 byte): The number of characters in the checksum for the MAT (fixed at 4)
- Checksum for the MAT (4 bytes): Result of checksum calculation for the user boot MAT: the total of all data in the MAT, in byte units.
- SUM (1 byte): Checksum (for the transmitted data)

(12) Sum Checking of the User MAT

In response to the command for sum checking of the user MAT, the boot program adds all bytes of data in the user MAT and returns the result.

Command	H'4B

- Command H'4B (1 byte): Sum checking of the user MAT

Response	H'5B	Size	Checksum for the MAT	SUM

- Response H'5B (1 byte): Response to sum checking of the user MAT
- Size (1 byte): The number of characters in the checksum for the MAT (fixed at 4)
- Checksum for the MAT (4 bytes): Result of checksum calculation for the user MAT: the total of all data in the MAT, in byte units.
- SUM (1 byte): Checksum (for the transmitted data)

(13) Blank Checking of the User Boot MAT

In response to the command for blank checking of the user boot MAT, the boot program checks to see if the whole of the user boot MAT is blank; the value returned indicates the result.

Command	H'4C

- Command H'4C (1 byte): Blank checking of the user boot MAT

Response	H'06

- Response H'06 (1 byte): Response to blank checking of the user boot MAT
  This ACK code is returned when the whole area is blank (all bytes are H'FF).

Error response	H'CC	H'52

- Error response H'CC (1 byte): Error response to blank checking of the user boot MAT
- Error code H'52 (1 byte): Non-erased error

(14) Blank Checking of the User MAT

In response to the command for blank checking of the user MAT, the boot program checks to see if the whole of the user MAT is blank; the value returned indicates the result.

Command	H'4D

- Command H'4D (1 byte): Blank checking of the user boot MAT

Response	H'06

- Response H'06 (1 byte): Response to blank checking of the user MAT
  The ACK code is returned when the whole area is blank (all bytes are H'FF).

Error response	H'CD	H'52

- Error response H'CD (1 byte): Error response to blank checking of the user MAT
- Error code H'52 (1 byte): Non-erased error

(15) Inquiry on Boot Program State

In response to the command for inquiry on the state of the boot program, the boot program returns an indicator of its current state and error information. This inquiry can be made in the inquiry-and-selection state or the programming/erasure state.

Command	H'4F

- Command H'4F (1 byte): Inquiry on boot program state

Response	H'5F	Size	STATUS	ERROR	SUM

- Response H'5F (1 byte): Response to the inquiry regarding boot-program state
- Size (1 byte): The number of characters in STATUS and ERROR (fixed at 2)
- STATUS (1 byte): State of the standard boot program
  See table 21.14, Status Codes.
- ERROR (1 byte): Error state (indicates whether the program is in normal operation or an error has occurred)
  ERROR = 0: Normal
  ERROR ≠ 0: Error
  See table 21.15, Error Codes.
- SUM (1 byte): Checksum

Table 21.14 Status Codes

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 status (bit-rate selection complete)
H'31	Erasing the user MAT or user boot MAT
H'3F	Waiting for programming/erasure selection (erasure complete)
H'4F	Waiting to receive data for programming (programming complete)
H'5F	Waiting for erasure block specification (erasure complete)

Table 21.15 Error Codes

Code	Description
H'00	No error
H'11	Sum check error
H'21	Non-matching device code error
H'22	Non-matching clock mode error
H'24	Bit-rate selection failure
H'25	Input frequency error
H'26	Frequency multiplier error
H'27	Operating frequency error
H'29	Block number error
H'2A	Address error
H'2B	Data length error (size error)
H'51	Erasure error
H'52	Non-erased error
H'53	Programming error
H'54	Selection processing error
H'80	Command error
H'FF	Bit-rate matching acknowledge error

21.8.2 Areas for Storage of the Procedural Program and Data for Programming

In the descriptions in the previous section, storable areas for the programming/erasing procedure programs and program data are assumed to be in on-chip RAM. However, the procedure programs and data can be stored in and executed from other areas (e.g. external address space) as long as the following conditions are satisfied.

1. The on-chip programming/erasing program is downloaded from the address set by FTDAR in on-chip RAM, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use 128 bytes or more as a stack. Make sure this area is reserved.
3. Since download by setting the SCO bit to 1 will cause the MATs to be switched, it should be executed in on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been decided.

5. The flash memory is not accessible during programming/erasing operations. Therefore, the programming/erasing program must be downloaded to on-chip RAM in advance. Areas for executing each procedure program for initiating programming/erasing, the user program at the user branch destination for programming/erasing, the interrupt vector table, and the interrupt processing routine must be located in on-chip memory other than flash memory or the external address space.
6. After programming/erasing, access to flash memory is inhibited until FKEY is cleared.

   A reset state ($\overline{RES}$ = 0) for more than at least 100 μs must be taken when the LSI mode is changed to reset on completion of a programming/erasing operation.

   Transitions to the reset state during programming/erasing are inhibited. When the reset signal is accidentally input to the LSI, a longer period in the reset state than usual (100 μs) is needed before the reset signal is released.
7. Switching of the MATs by FMATS is needed for programming/erasing of the user MAT in user boot mode. The program which switches the MATs should be executed from the on-chip RAM. For details, see section 21.7.1, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching the MATs.
8. When the program data storage area indicated by the FMPDR parameter in the programming processing is within the flash memory area, an error will occur. Therefore, temporarily transfer the program data to on-chip RAM to change the address set in FMPDR to an address other than flash memory.

Based on these conditions, tables 21.16 and 21.17 show the areas in which the program data can be stored and executed according to the operation type and mode.

Table 21.16 Executable MAT

	Initiated Mode	
Operation	**User Program Mode**	**User Boot Mode***
Programming	Table 21.17 (1)	Table 21.17 (3)
Erasing	Table 21.17 (2)	Table 21.17 (4)

Note: * Programming/Erasing is possible to user MATs.

Table 21.17 (1) Usable Area for Programming in User Program Mode

		Storable/Executable Area			Selected MAT	
	Item	On-Chip RAM	User MAT	External Space	User MAT	Embedded Program Storage MAT
Programming procedure	Program data storage area	√	X*	√	—	—
	Selecting on-chip program to be downloaded	√	√	√	√	
	Writing H'A5 to key register	√	√	√	√	
	Writing 1 to SCO in FCCS (download)	√	X	X		√
	Key register clearing	√	√	√	√	
	Judging download result	√	√	√	√	
	Download error processing	√	√	√	√	
	Setting initialization parameters	√	√	√	√	
	Initialization	√	X	X	√	
	Judging initialization result	√	√	√	√	
	Initialization error processing	√	√	√	√	
	Interrupt processing routine	√	X	√	√	
	Writing H'5A to key register	√	√	√	√	
	Setting programming parameters	√	X	√	√	
	Programming	√	X	X	√	
	Judging programming result	√	X	√	√	
	Programming error processing	√	X	√	√	
	Key register clearing	√	X	√	√	

Note: * If the data has been transferred to on-chip RAM in advance, this area can be used.

Table 21.17 (2) Usable Area for Erasure in User Program Mode

		Storable/Executable Area			Selected MAT	
	Item	On-Chip RAM	User MAT	External Space	User MAT	Embedded Program Storage MAT
Erasing procedure	Selecting on-chip program to be downloaded	√	√	√	√	
	Writing H'A5 to key register	√	√	√	√	
	Writing 1 to SCO in FCCS (download)	√	X	X		√
	Key register clearing	√	√	√	√	
	Judging download result	√	√	√	√	
	Download error processing	√	√	√	√	
	Setting initialization parameters	√	√	√	√	
	Initialization	√	X	X	√	
	Judging initialization result	√	√	√	√	
	Initialization error processing	√	√	√	√	
	Interrupt processing routine	√	X	√	√	
	Writing H'5A to key register	√	√	√	√	
	Setting erasure parameters	√	X	√	√	
	Erasure	√	X	X	√	
	Judging erasure result	√	X	√	√	
	Erasing error processing	√	X	√	√	
	Key register clearing	√	X	√	√	

Table 21.17 (3) Usable Area for Programming in User Boot Mode

		Storable/Executable Area			Selected MAT		
	Item	On-Chip RAM	User Boot MAT	External Space	User MAT	User Boot MAT	Embedded Program Storage Area
Programming procedure	Program data storage area	√	X*[1]	√	—	—	—
	Selecting on-chip program to be downloaded	√	√	√		√	
	Writing H'A5 to key register	√	√	√		√	
	Writing 1 to SCO in FCCS (download)	√	X	X			√
	Key register clearing	√	√	√		√	
	Judging download result	√	√	√		√	
	Download error processing	√	√	√		√	
	Setting initialization parameters	√	√	√		√	
	Initialization	√	X	X		√	
	Judging initialization result	√	√	√		√	
	Initialization error processing	√	√	√		√	
	Interrupt processing routine	√	X	√		√	
	Switching MATs by FMATS	√	X	X	√		
	Writing H'5A to Key Register	√	X	√	√		

Table 21.17 (3) Usable Area for Programming in User Boot Mode (cont)

		Storable/Executable Area			Selected MAT		
	Item	On-Chip RAM	User Boot MAT	External Space	User MAT	User Boot MAT	Embedded Program Storage Area
Programming procedure	Setting programming parameters	√	X	√	√		
	Programming	√	X	X	√		
	Judging programming result	√	X	√	√		
	Programming error processing	√	X*2	√	√		
	Key register clearing	√	X	√	√		
	Switching MATs by FMATS	√	X	X		√	

Notes: 1. If the data has been transferred to on-chip RAM in advance, this area can be used.
2. If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

Table 21.17 (4) Usable Area for Erasure in User Boot Mode

		Storable/Executable Area			Selected MAT		
	Item	On-Chip RAM	User Boot MAT	External Space	User MAT	User Boot MAT	Embedded Program Storage Area
Erasing procedure	Selecting on-chip program to be downloaded	√	√	√		√	
	Writing H'A5 to key register	√	√	√		√	
	Writing 1 to SCO in FCCS (download)	√	X	X			√
	Key register clearing	√	√	√		√	
	Judging download result	√	√	√		√	
	Download error processing	√	√	√		√	
	Setting initialization parameters	√	√	√		√	
	Initialization	√	X	X		√	
	Judging initialization result	√	√	√		√	
	Initialization error processing	√	√	√		√	
	Interrupt processing routine	√	X	√		√	
	Switching MATs by FMATS	√	X	X		√	
	Writing H'5A to key register	√	X	√	√		
	Setting erasure parameters	√	X	√	√		

Table 21.17 (4) Usable Area for Erasure in User Boot Mode (cont)

		Storable/Executable Area			Selected MAT		
	Item	On-Chip RAM	User Boot MAT	External Space	User MAT	User Boot MAT	Embedded Program Storage Area
Erasing procedure	Erasure	√	X	X	√		
	Judging erasure result	√	X	√	√		
	Erasing error processing	√	X*	√	√		
	Key register clearing	√	X	√	√		
	Switching MATs by FMATS	√	X	X		√	

Note: * If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

21.9 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 Renesas 512-kbyte flash memory on-chip MCU device type (F-ZTAT512DV3_15A).

Section 22 On-Chip RAM

This LSI has an on-chip RAM module which can be used to store instructions or data.

On-chip RAM operation and write access to the RAM can be enabled or disabled through the RAM enable bits and RAM write enable bits.

22.1 Features

- Pages

  The on-chip RAM is divided into four pages (pages 0 to 3).
- Memory map

  The on-chip RAM is located in the address spaces shown in table 22.1.

Table 22.1 On-Chip RAM Address Spaces

Page	Address
Page 0	H'FFF80000 to H'FFF81FFF
Page 1	H'FFF82000 to H'FFF83FFF
Page 2	H'FFF84000 to H'FFF85FFF
Page 3	H'FFF86000 to H'FFF87FFF

- Ports

  Each page has two independent read and write ports and is connected to the internal bus (I bus), CPU instruction fetch bus (F bus), and CPU memory access bus (M bus). (Note that the F bus is connected only to the read ports.)

  The F bus and M bus are used for access by the CPU, and the I bus is used for access by the DMAC.
- Priority

  When the same page is accessed from different buses simultaneously, the access is processed according to the priority. The priority is I bus > M bus > F bus.

22.2 Usage Notes

22.2.1 Page Conflict

When the same page is accessed from different buses simultaneously, a conflict on the page occurs. Although each access is completed correctly, this kind of conflict degrades the memory access speed. Therefore, it is advisable to provide software measures to prevent such conflicts as far as possible. For example, no conflict will arise if different memory or pages are accessed by each bus.

22.2.2 RAME and RAMWE Bits

Before disabling memory operation or write access through the RAME or RAMWE bit, be sure to read from any address and then write to the same address in each page; otherwise, the last written data in each page may not be actually written to the RAM. For setting the RAME and RAMWE bits, see section 23.3.5, System Control Register 1 (SYSCR1), and section 23.3.6, System Control Register 2 (SYSCR2).

```
// For page 0
  MOV.L  #H'FFF80000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 1
  MOV.L  #H'FFF82000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 2
  MOV.L  #H'FFF84000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For page 3
  MOV.L  #H'FFF86000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0
```

Figure 22.1 Examples of Read/Write before Disabling RAM

Section 23 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.

23.1 Features

23.1.1 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 23.1 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 23.1 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 (RAM) Halts (Flash memory)	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, and STBCR4 to 1	Runs	Runs	Held	Specified module halts (contents are held)	Specified module halts	Auto-refreshing	• Clear MSTP bit to 0 • Power-on reset (only for H-UDI, UBC, and DMAC)

Note: * The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

23.1.2 Reset

A reset is used when the power is turned on or to run the LSI again from the initialized state. There are two types of reset: power-on reset and manual reset. In a power-on reset, all the ongoing processing is halted and any unprocessed events are canceled, and the reset processing starts immediately. On the other hand, a manual reset does not interrupt processing to retain external memory data. Conditions for generating a power-on reset or manual reset are as follows:

(1) Power-on Reset

1. A low level is input to the $\overline{\text{RES}}$ pin.
2. The watchdog timer (WDT) starts counting with the WT/$\overline{\text{IT}}$ bit in WTCSR set to 1 and with the RSTS bit in WRCSR set to 0 while the RSRE bit in WRCSR is 1, and the counter overflows.
3. The H-UDI reset is generated (for details on the H-UDI reset, see section 24, User Debugging Interface (H-UDI)).

(2) Manual Reset

1. A low level is input to the $\overline{\text{MRES}}$ pin.
2. The WDT starts counting with the WT/$\overline{\text{IT}}$ bit in WTCSR set to 1 and with the RSTS bit in WRCSR set to 1 while the RSRE bit in WRCSR is 1, and the counter overflows.

23.2 Input/Output Pins

Table 23.2 lists the pins used for power-down modes.

Table 23.2 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.

23.3 Register Descriptions

The following registers are used in power-down modes.

Table 23.3 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'F4	H'FFFE040C	8
System control register 1	SYSCR1	R/W	H'FF	H'FFFE0402	8
System control register 2	SYSCR2	R/W	H'FF	H'FFFE0404	8

23.3.1 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.

23.3.2 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	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	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 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

23.3.3 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	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	MSTP31	MSTP30
Initial value:	0	1	1	1	1	1	1	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	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	MSTP34	1	R/W	Module Stop 34 When the MSTP34 bit is set to 1, the supply of the clock to the POE2 is halted. 0: POE2 runs. 1: Clock supply to POE2 halted.
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the supply of the clock to the IIC3 is halted. 0: IIC3 runs. 1: Clock supply to IIC3 halted.
2	MSTP32	1	R/W	Module Stop 32 When the MSTP32 bit is set to 1, the supply of the clock to the ADC is halted. 0: ADC runs. 1: Clock supply to ADC halted.
1	MSTP31	1	R/W	Module Stop 31 When the MSTP31 bit is set to 1, the supply of the clock to the DAC is halted. 0: DAC runs. 1: Clock supply to DAC halted.
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: Flash memory runs. 1: Clock supply to flash memory halted.

23.3.4 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'F4 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
	MSTP47	MSTP46	MSTP45	MSTP44	-	MSTP42	MSTP41	-
Initial value:	1	1	1	1	0	1	1	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP47	1	R/W	Module Stop 47 When the MSTP47 bit is set to 1, the supply of the clock to the SCIF0 is halted. 0: SCIF0 runs. 1: Clock supply to SCIF0 halted.
6	MSTP46	1	R/W	Module Stop 46 When the MSTP46 bit is set to 1, the supply of the clock to the SCIF1 is halted. 0: SCIF1 runs. 1: Clock supply to SCIF1 halted.
5	MSTP45	1	R/W	Module Stop 45 When the MSTP45 bit is set to 1, the supply of the clock to the SCIF2 is halted. 0: SCIF2 runs. 1: Clock supply to SCIF2 halted.
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.

Bit	Bit Name	Initial Value	R/W	Description
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	MSTP41	1	R/W	Module Stop 41 When the MSTP41 bit is set to 1, the supply of the clock to the WAVEIF is halted. 0: WAVEIF runs. 1: Clock supply to WAVEIF halted.
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

23.3.5 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 possible.

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.

To enable the on-chip RAM by setting the RAME bit to 1, place an instruction to read data from SYSCR1 immediately after an instruction to write to SYSCR1. If an instruction to access the on-chip RAM is placed immediately after the instruction to write to SYSCR1, normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAME3	RAME2	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
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 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	RAME3	1	R/W	RAM Enable 3 (corresponding RAM addresses: H'FFF86000 to H'FFF87FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RAME2	1	R/W	RAM Enable 2 (corresponding RAM addresses: H'FFF84000 to H'FFF85FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
1	RAME1	1	R/W	RAM Enable 1 (corresponding RAM addresses: H'FFF82000 to H'FFF83FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
0	RAME0	1	R/W	RAM Enable 0 (corresponding RAM addresses: H'FFF80000 to H'FFF81FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled

23.3.6 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 RAMWE 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.

To enable the on-chip RAM by setting the RAMWE bit to 1, locate an instruction to read data from SYSCR2 immediately after an instruction to write to SYSCR2. If an instruction to access the on-chip RAM is located immediately after the instruction to write to SYSCR2, normal access is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	-	-	-	-	RAM WE3	RAM WE2	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
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 1	R	Reserved These bits are always read as 1. The write value should always be 1.
3	RAMWE3	1	R/W	RAM Write Enable 3 (corresponding RAM addresses: H'FFF86000 to H'FFF87FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled

Bit	Bit Name	Initial Value	R/W	Description
2	RAMWE2	1	R/W	RAM Write Enable 2 (corresponding RAM addresses: H'FFF84000 to H'FFF85FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
1	RAMWE1	1	R/W	RAM Write Enable 1 (corresponding RAM addresses: H'FFF82000 to H'FFF83FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled
0	RAMWE0	1	R/W	RAM Write Enable 0 (corresponding RAM addresses: H'FFF80000 to H'FFF81FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled

23.4 Operation

23.4.1 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 DMA address error
  When a DMA address error occurs, sleep mode is canceled and DMA address error exception handling is executed.
- Canceling with a reset
  Sleep mode is canceled by a power-on reset or a manual reset.

23.4.2 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 remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. Table 23.4 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 23.4 Register States in Software Standby Mode

Module Name	Initialized Registers	Registers Whose Content is Retained
Interrupt controller (INTC)	—	All registers
Clock pulse generator (CPG)	—	All registers
User break controller (UBC)	—	All registers
Bus state controller (BSC)	—	All registers
A/D converter (ADC)	All registers	—
I/O port	—	All registers
User debugging interface (H-UDI)	—	All registers
Serial communication interface with FIFO (SCIF)	—	All registers
Direct memory access controller (DMAC)	—	All registers
Multi-function timer pulse unit 2 (MTU2)	—	All registers
Multi-function timer pulse unit 2S (MTU2S)	—	All registers
Port output enable 2 (POE2)	—	All registers
Compare match timer (CMT)	All registers	—
I^2C bus interface 3 (IIC3)	BC2 and BC0 bits in ICMR register	Other than BC[2:0] bits in ICMR
D/A converter (DAC)	—	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 or IRQ) or a reset (manual reset or power-on reset).

(a) Exit from Software Standby by an Interrupt

When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) of the interrupt controller (INTC)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)) is detected, clock oscillation is started. This clock pulse is supplied only to the oscillation settling counter (WDT) used to count the oscillation settling time.

After the elapse of the time set in the clock select bits (CKS[2:0]) in the watchdog timer control/status register (WTCSR) of the WDT before the transition to software standby mode, the WDT overflow occurs. Since this overflow indicates that the clock has been stabilized, the clock pulse will be supplied to the entire chip after this overflow. Software standby mode is cleared and NMI interrupt exception handling (IRQ interrupt exception handling in the case of IRRQ) starts.

When canceling software standby mode by the NMI interrupt or IRQ interrupt, set the CKS[2:0] bits so that the WDT overflow period will be equal to or longer than the oscillation settling time.

The clock output phase of the CK pin may be unstable immediately after detecting an interrupt and until software standby mode is canceled. When software standby mode is canceled 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 the CPU returns from software standby mode (when the clock is initiated after the oscillation settling). When software standby mode is canceled 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 the CPU returns from software standby mode (when the clock is initiated after the oscillation settling) (This is the same with the IRQ pin.)

(b) 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 or manual reset state, 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.

23.4.3 Software Standby Mode Application Example

This example describes a transition to software standby mode on the falling edge of the NMI signal, and cancellation on the rising edge of the NMI signal. The timing is shown in figure 23.1.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in ICR is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by the NMI exception service routine, the STBY bit in STBCR is set to 1, and a SLEEP instruction is executed, software standby mode is entered. Thereafter, software standby mode is canceled when the NMI pin is changed from low to high level.

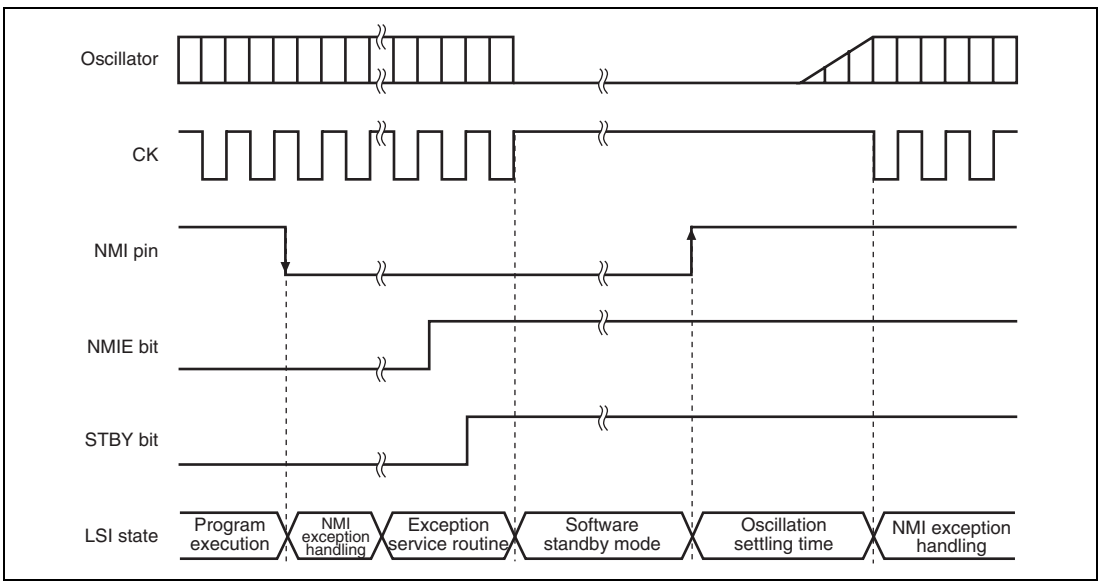


Figure 23.1 NMI Timing in Software Standby Mode (Application Example)

23.4.4 Module Standby Function

(1) Transition to Module Standby Function

Setting the standby control register MSTP bits to 1 halts the supply of clocks to the corresponding on-chip peripheral modules. This function can be used to reduce the power consumption in normal mode and sleep mode. Disable a module before placing it in module standby mode. In addition, do not access the module's registers while it is in the module standby state.

The register states are the same as those in software standby mode. For details of register states, see table 23.4.

However, the states of the CMT and DAC registers are exceptional. In the CMT, all registers are initialized in software standby mode, but retain their previous values in module standby mode. In the DAC, all registers retain their previous values in software standby mode, but are initialized in module standby mode.

(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, and DMAC). 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 24 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for emulator support.

24.1 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 24.1 shows a block diagram of the H-UDI.

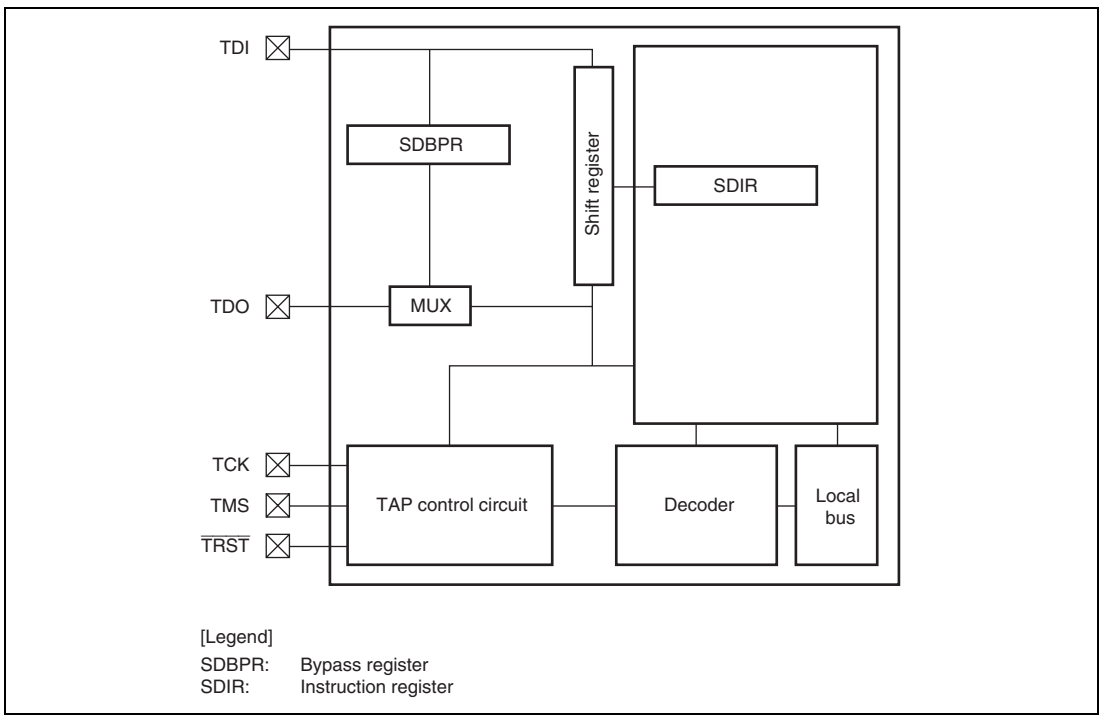


Figure 24.1 Block Diagram of H-UDI

24.2 Input/Output Pins

Table 24.1 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 24.2.
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 24.4.2, 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 24.4.3, TDO Output Timing, for more information.
ASE mode select pin	$\overline{\text{ASEMD}}$*	Input	If a low level is input at the $\overline{\text{ASEMD}}$ 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 $\overline{\text{ASEMD}}$ 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.

24.3 Register Descriptions

The H-UDI has the following registers.

Table 24.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	SDBPR	—	—	—	—
Instruction register	SDIR	R	H'EFFD	H'FFFE2000	16

24.3.1 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.

24.3.2 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 24.3.
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.

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.

Table 24.3 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

24.4 Operation

24.4.1 TAP Controller

Figure 24.2 shows the internal states of the TAP controller.

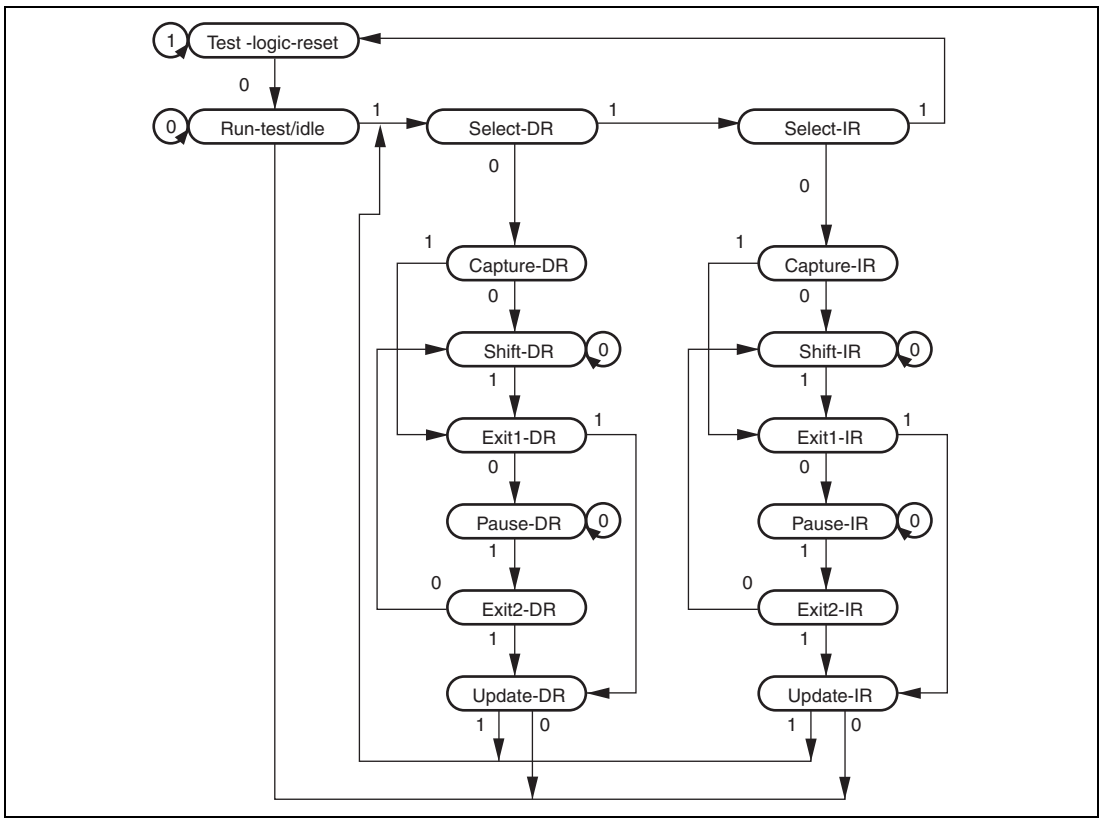


Figure 24.2 TAP Controller State Transitions

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 24.4.3, TDO Output Timing. The TDO is at high impedance, except with shift-DR and shift-IR states. During the change to $\overline{\text{TRST}}$ = 0, there is a transition to test-logic-reset asynchronously with TCK.

24.4.2 Reset Configuration

Table 24.4 Reset Configuration

$\overline{\text{ASEMD}}$*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
		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
$\overline{\text{ASEMD}}$ = H, normal mode
$\overline{\text{ASEMD}}$ = 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.

24.4.3 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.

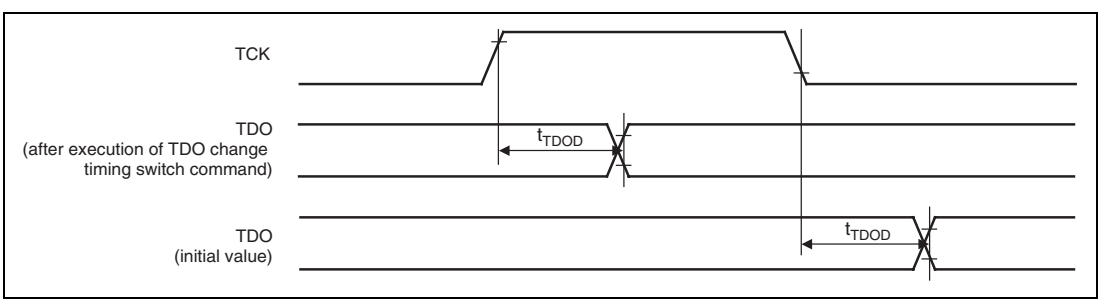


Figure 24.3 H-UDI Data Transfer Timing

24.4.4 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.

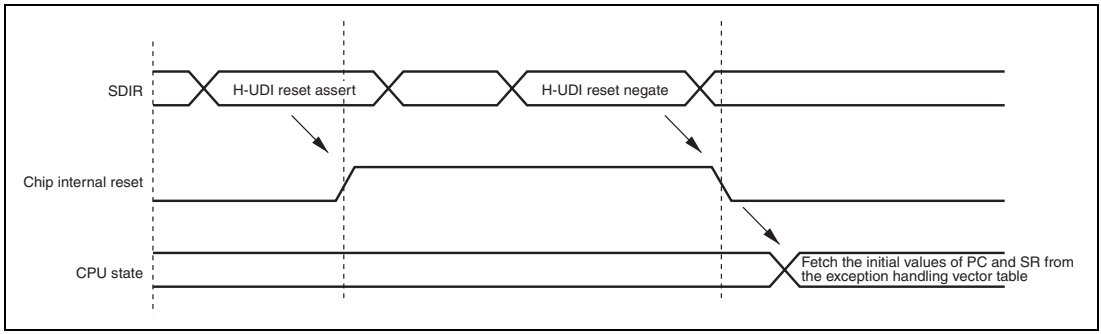


Figure 24.4 H-UDI Reset

24.4.5 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.

24.5 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 and H-UDI module standby state, all of the functions in the H-UDI cannot be used. To retain the TAP status before and after standby mode, keep TCK high before entering standby mode.

Section 25 WAVE Interface (WAVEIF)

The WAVE interface (WAVEIF) is the interface function of Myway Labs Real-time CPU Scope WAVE™. For more information on WAVE™, contact Myway Labs Co., Ltd.

25.1 Features

Conforms to WAVE1.0 Level C

25.2 Input/Output Pins

Table 25.1 Pin Configuration

Pin Name	Symbol	I/O	Function
WAVE clock pin	WSCK	Output	WAVE interface clock output
WAVE receive data	WRXD	Input	WAVE interface receive data input
WAVE transmit data	WTXD	Output	WAVE interface transmit data output

Section 26 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.

26.1 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	16	H'FFFE0414	8
INTC	Interrupt control register 0	ICR0	16	H'FFFE0800	16, 32
	Interrupt control register 1	ICR1	16	H'FFFE0802	16, 32
	IRQ interrupt request register	IRQRR	16	H'FFFE0806	16, 32
	Bank control register	IBCR	16	H'FFFE080C	16, 32
	Bank number register	IBNR	16	H'FFFE080E	16, 32
	Interrupt priority register 01	IPR01	16	H'FFFE0818	16, 32
	Interrupt priority register 02	IPR02	16	H'FFFE081A	16, 32
	Interrupt priority register 05	IPR05	16	H'FFFE0820	16, 32
	Interrupt priority register 06	IPR06	16	H'FFFE0C00	16, 32
	Interrupt priority register 07	IPR07	16	H'FFFE0C02	16, 32
	Interrupt priority register 08	IPR08	16	H'FFFE0C04	16, 32
	Interrupt priority register 09	IPR09	16	H'FFFE0C06	16, 32
	Interrupt priority register 10	IPR10	16	H'FFFE0C08	16, 32
	Interrupt priority register 11	IPR11	16	H'FFFE0C0A	16, 32
	Interrupt priority register 12	IPR12	16	H'FFFE0C0C	16, 32
	Interrupt priority register 13	IPR13	16	H'FFFE0C0E	16, 32
	Interrupt priority register 14	IPR14	16	H'FFFE0C10	16, 32
	Interrupt priority register 15	IPR15	16	H'FFFE0C12	16, 32
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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
UBC	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
BSC	Common control register	CMNCR	32	H'FFFC0000	32
	CS0 space bus control register	CS0BCR	32	H'FFFC0004	32
	CS1 space bus control register	CS1BCR	32	H'FFFC0008	32
	CS2 space bus control register	CS2BCR	32	H'FFFC000C	32
	CS3 space bus control register	CS3BCR	32	H'FFFC0010	32
	CS4 space bus control register	CS4BCR	32	H'FFFC0014	32
	CS5 space bus control register	CS5BCR	32	H'FFFC0018	32
	CS6 space bus control register	CS6BCR	32	H'FFFC001C	32
	CS7 space bus control register	CS7BCR	32	H'FFFC0020	32
	CS0 space wait control register	CS0WCR	32	H'FFFC0028	32
	CS1 space wait control register	CS1WCR	32	H'FFFC002C	32
	CS2 space wait control register	CS2WCR	32	H'FFFC0030	32
	CS3 space wait control register	CS3WCR	32	H'FFFC0034	32
	CS4 space wait control register	CS4WCR	32	H'FFFC0038	32
	CS5 space wait control register	CS5WCR	32	H'FFFC003C	32
	CS6 space wait control register	CS6WCR	32	H'FFFC0040	32
	CS7 space wait control register	CS7WCR	32	H'FFFC0044	32
	SDRAM control register	SDCR	32	H'FFFC004C	32
	Refresh timer control/status register	RTCSR	16	H'FFFC0050	32
	Refresh timer counter	RTCNT	16	H'FFFC0054	32
	Refresh time constant register	RTCOR	16	H'FFFC0058	32
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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	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
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
DMAC	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
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer control register_0	TCR_0	8	H'FFFE4300	8
	Timer mode register_0	TMDR_0	8	H'FFFE4301	8
	Timer I/O control register H_0	TIORH_0	8	H'FFFE4302	8
	Timer I/O control register L_0	TIORL_0	8	H'FFFE4303	8
	Timer interrupt enable register_0	TIER_0	8	H'FFFE4304	8
	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
	Timer general register B_0	TGRB_0	16	H'FFFE430A	16
	Timer general register C_0	TGRC_0	16	H'FFFE430C	16
	Timer general register D_0	TGRD_0	16	H'FFFE430E	16
	Timer general register E_0	TGRE_0	16	H'FFFE4320	16
	Timer general register F_0	TGRF_0	16	H'FFFE4322	16
	Timer interrupt enable register2_0	TIER2_0	8	H'FFFE4324	8
	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
	Timer mode register_1	TMDR_1	8	H'FFFE4381	8
	Timer I/O control register_1	TIOR_1	8	H'FFFE4382	8
	Timer interrupt enable register_1	TIER_1	8	H'FFFE4384	8
	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
	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
	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
	Timer status register_2	TSR_2	8	H'FFFE4005	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer counter_2	TCNT_2	16	H'FFFE4006	16
	Timer general register A_2	TGRA_2	16	H'FFFE4008	16
	Timer general register B_2	TGRB_2	16	H'FFFE400A	16
	Timer control register_3	TCR_3	8	H'FFFE4200	8
	Timer mode register_3	TMDR_3	8	H'FFFE4202	8
	Timer I/O control register H_3	TIORH_3	8	H'FFFE4204	8
	Timer I/O control register L_3	TIORL_3	8	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	8	H'FFFE4208	8
	Timer status register_3	TSR_3	8	H'FFFE422C	8
	Timer counter_3	TCNT_3	16	H'FFFE4210	16
	Timer general register A_3	TGRA_3	16	H'FFFE4218	16
	Timer general register B_3	TGRB_3	16	H'FFFE421A	16
	Timer general register C_3	TGRC_3	16	H'FFFE4224	16
	Timer general register D_3	TGRD_3	16	H'FFFE4226	16
	Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238	8
	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
	Timer I/O control register L_4	TIORL_4	8	H'FFFE4207	8
	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
	Timer general register B_4	TGRB_4	16	H'FFFE421E	16
	Timer general register C_4	TGRC_4	16	H'FFFE4228	16
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer A/D converter start request cycle set register A_4	TADCORA_4	16	H'FFFE4244	16
	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
	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
	Timer counter V_5	TCNTV_5	16	H'FFFE4090	16
	Timer counter W_5	TCNTW_5	16	H'FFFE40A0	16
	Timer general register U_5	TGRU_5	16	H'FFFE4082	16
	Timer general register V_5	TGRV_5	16	H'FFFE4092	16
	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
	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
	Timer output control register 2	TOCR2	8	H'FFFE420F	8
	Timer gate control register	TGCR	8	H'FFFE420D	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2	Timer cycle control register	TCDR	16	H'FFFE4214	16
	Timer dead time data register	TDDR	16	H'FFFE4216	16
	Timer subcounter	TCNTS	16	H'FFFE4220	16
	Timer cycle buffer register	TCBR	16	H'FFFE4222	16
	Timer interrupt skipping set register	TITCR	8	H'FFFE4230	8
	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 synchronous clear register	TSYCR	8	H'FFFE4250	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
	Timer mode register_3S	TMDR_3S	8	H'FFFE4A02	8
	Timer I/O control register H_3S	TIORH_3S	8	H'FFFE4A04	8
	Timer I/O control register L_3S	TIORL_3S	8	H'FFFE4A05	8
	Timer interrupt enable register_3S	TIER_3S	8	H'FFFE4A08	8
	Timer status register_3S	TSR_3S	8	H'FFFE4A2C	8
	Timer counter_3S	TCNT_3S	16	H'FFFE4A10	16
	Timer general register A_3S	TGRA_3S	16	H'FFFE4A18	16
	Timer general register B_3S	TGRB_3S	16	H'FFFE4A1A	16
	Timer general register C_3S	TGRC_3S	16	H'FFFE4A24	16
	Timer general register D_3S	TGRD_3S	16	H'FFFE4A26	16
	Timer buffer operation transfer mode register_3S	TBTM_3S	8	H'FFFE4A38	8
	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
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2S	Timer general register A_4S	TGRA_4S	16	H'FFFE4A1C	16
	Timer general register B_4S	TGRB_4S	16	H'FFFE4A1E	16
	Timer general register C_4S	TGRC_4S	16	H'FFFE4A28	16
	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
	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
	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
	Timer start register_5S	TSTR_5S	8	H'FFFE48B4	8
	Timer counter U_5S	TCNTU_5S	16	H'FFFE4880	16
	Timer counter V_5S	TCNTV_5S	16	H'FFFE4890	16
	Timer counter W_5S	TCNTW_5S	16	H'FFFE48A0	16
	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

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
MTU2S	Timer synchronous register S	TSYRS	8	H'FFFE4A81	8
	Timer counter synchronous start register S	TRWERS	8	H'FFFE4A84	8
	Timer read/write enable register S	TOERS	8	H'FFFE4A0A	8
	Timer output control register 1S	TOCR1S	8	H'FFFE4A0E	8
	Timer output control register 2S	TOCR2S	8	H'FFFE4A0F	8
	Timer gate control register S	TGCRS	8	H'FFFE4A0D	8
	Timer cycle control register S	TCDRS	16	H'FFFE4A14	16
	Timer dead time data register S	TDDRS	16	H'FFFE4A16	16
	Timer subcounter S	TCNTSS	16	H'FFFE4A20	16
	Timer cycle buffer register S	TCBRS	16	H'FFFE4A22	16
	Timer interrupt skipping set register S	TITCRS	8	H'FFFE4A30	8
	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
	Port output enable control register 1	POECR1	8	H'FFFE500B	8
	Port output enable control register 2	POECR2	16	H'FFFE500C	16
CMT	Compare match timer start register	CMSTR	16	H'FFFEC000	16
	Compare match timer control/status register_0	CMCSR_0	16	H'FFFEC002	16
	Compare match counter_0	CMCNT_0	16	H'FFFEC004	16
	Compare match constant register_0	CMCOR_0	16	H'FFFEC006	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
CMT	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	*
	Watchdog timer counter	WTCNT	16	H'FFFE0002	*
	Watchdog reset control/status register	WRCSR	16	H'FFFE0004	*
SCIF	Serial mode register_0	SCSMR_0	16	H'FFFE8000	16
	Bit rate register_0	SCBRR_0	8	H'FFFE8004	8
	Serial control register_0	SCSCR_0	16	H'FFFE8008	16
	Transmit FIFO data register_0	SCFTDR_0	8	H'FFFE800C	8
	Serial status register_0	SCFSR_0	16	H'FFFE8010	16
	Receive FIFO data register_0	SCFRDR_0	8	H'FFFE8014	8
	FIFO control register_0	SCFCR_0	16	H'FFFE8018	16
	FIFO data count register_0	SCFDR_0	16	H'FFFE801C	16
	Serial port register_0	SCSPTR_0	16	H'FFFE8020	16
	Line status register_0	SCLSR_0	16	H'FFFE8024	16
	Serial mode register_1	SCSMR_1	16	H'FFFE8800	16
	Bit rate register_1	SCBRR_1	8	H'FFFE8804	8
	Serial control register_1	SCSCR_1	16	H'FFFE8808	16
	Transmit FIFO data register_1	SCFTDR_1	8	H'FFFE880C	8
	Serial status register_1	SCFSR_1	16	H'FFFE8810	16
	Receive FIFO data register_1	SCFRDR_1	8	H'FFFE8814	8
	FIFO control register_1	SCFCR_1	16	H'FFFE8818	16
	FIFO data count register_1	SCFDR_1	16	H'FFFE881C	16
	Serial port register_1	SCSPTR_1	16	H'FFFE8820	16
	Line status register_1	SCLSR_1	16	H'FFFE8824	16
	Serial extended mode register_1	SCSEMR_1	8	H'FFFE8900	8
	Serial mode register_2	SCSMR_2	16	H'FFFE9000	16
	Bit rate register_2	SCBRR_2	8	H'FFFE9004	8
	Serial control register_2	SCSCR_2	16	H'FFFE9008	16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
SCIF	Transmit FIFO data register_2	SCFTDR_2	8	H'FFFE900C	8
	Serial status register_2	SCFSR_2	16	H'FFFE9010	16
	Receive FIFO data register_2	SCFRDR_2	8	H'FFFE9014	8
	FIFO control register_2	SCFCR_2	16	H'FFFE9018	16
	FIFO data count register_2	SCFDR_2	16	H'FFFE901C	16
	Serial port register_2	SCSPTR_2	16	H'FFFE9020	16
	Line status register_2	SCLSR_2	16	H'FFFE9024	16
	Serial extended mode register_2	SCSEMR_2	8	H'FFFE9100	8
	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
IIC3	I^2C bus control register 1	ICCR1	8	H'FFFEE000	8
	I^2C bus control register 2	ICCR2	8	H'FFFEE001	8
	I^2C bus mode register	ICMR	8	H'FFFEE002	8
	I^2C bus interrupt enable register	ICIER	8	H'FFFEE003	8
	I^2C bus status register	ICSR	8	H'FFFEE004	8
	Slave address register	SAR	8	H'FFFEE005	8
	I^2C bus transmit data register	ICDRT	8	H'FFFEE006	8
	I^2C bus receive data register	ICDRR	8	H'FFFEE007	8
	NF2CYC register	NF2CYC	8	H'FFFEE008	8
ADC	A/D control register	ADCR	16	H'FFFFE800	8
	A/D status register	ADSR	16	H'FFFFE802	8
	A/D start trigger select register	ADSTRGR	8	H'FFFFE81C	8
	A/D analog input channel select register	ADANSR	8	H'FFFFE820	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
ADC	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 data register 4	ADDR4	16	H'FFFFE848	16
	A/D data register 5	ADDR5	16	H'FFFFE84A	16
	A/D data register 6	ADDR6	16	H'FFFFE84C	16
	A/D data register 7	ADDR7	16	H'FFFFE84E	16
DAC	D/A data register 0	DADR0	8	H'FFFE6800	8, 16
	D/A data register 1	DADR1	8	H'FFFE6801	8, 16
	D/A control register	DACR	8	H'FFFE6802	8, 16
PFC	Port A I/O register H	PAIORH	16	H'FFFE3804	8, 16, 32
	Port A I/O register L	PAIORL	16	H'FFFE3806	8, 16
	Port A control register H3	PACRH3	16	H'FFFE380A	8, 16
	Port A control register H2	PACRH2	16	H'FFFE380C	8, 16, 32
	Port A control register H1	PACRH1	16	H'FFFE380E	8, 16
	Port A control register L4	PACRL4	16	H'FFFE3810	8, 16, 32
	Port A control register L3	PACRL3	16	H'FFFE3812	8, 16
	Port A control register L2	PACRL2	16	H'FFFE3814	8, 16, 32
	Port A control register L1	PACRL1	16	H'FFFE3816	8, 16
	Port B I/O register H	PBIORH	16	H'FFFE3884	8, 16, 32
	Port B I/O register L	PBIORL	16	H'FFFE3886	8, 16
	Port B control register H4	PBCRH4	16	H'FFFE3888	8, 16, 32
	Port B control register H3	PBCRH3	16	H'FFFE388A	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 L4	PBCRL4	16	H'FFFE3890	8, 16, 32
	Port B control register L3	PBCRL3	16	H'FFFE3892	8, 16
	Port B control register L2	PBCRL2	16	H'FFFE3894	8, 16, 32
	Port B control register L1	PBCRL1	16	H'FFFE3896	8, 16

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port D I/O register	PDIOR	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 F control register L1	PFCRL1	16	H'FFFE3A96	8, 16
	IRQOUT function control register	IFCR	16	H'FFFE38A2	16
	WAVE function control register 2	WAVECR2	16	H'FFFE3A14	8, 16, 32
	WAVE function control register 1	WAVECR1	16	H'FFFE3A16	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 D data register L	PDDRL	16	H'FFFE3982	8, 16
	Port D port register L	PDPRL	16	H'FFFE399E	8, 16
	Port F data register	PFDR	16	H'FFFE3A82	8, 16
FLASH	Flash code control and status register	FCCS	8	H'8000C000	8
	Flash program code select register	FPCS	8	H'8000C001	8
	Flash erase code select register	FECS	8	H'8000C002	8
	Flash key code register	FKEY	8	H'8000C004	8
	Flash MAT select register	FMATS	8	H'8000C005	8
	Flash transfer destination address register	FTDAR	8	H'8000C006	8

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
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
H-UDI	Instruction register	SDIR	16	H'FFFE2000	16

Note: * The access sizes of the WDT registers are different between the read and write to prevent incorrect writing.

26.2 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	—	—	—	CKOEN	—	—	STC[1:0]	
		—	IFC[2:0]			RNGS	PFC[2:0]		
	MCLKCR	MSSCS[1:0]		—	—	—	—	MSDIVS[1:0]	
	ACLKCR	ASSCS[1:0]		—	—	—	—	ASDIVS[1:0]	
INTC	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE[1:0]		BOVE	—	—	—	—	—
		—	—	—	—	BN[3:0]			
	IPR01	IRQ0[3:0]				IRQ1[3:0]			
		IRQ2[3:0]				IRQ3[3:0]			
	IPR02	IRQ4[3:0]				IRQ5[3:0]			
		IRQ6[3:0]				IRQ7[3:0]			
	IPR05	—	—	—	—	—	—	—	—
		ADI[3:0]				—	—	—	—
	IPR06	DMAC0[3:0]				DMAC1[3:0]			
		DMAC2[3:0]				DMAC3[3:0]			
	IPR07	DMAC4[3:0]				DMAC5[3:0]			
		DMAC6[3:0]				DMAC7[3:0]			
	IPR08	CMT0[3:0]				CMT1[3:0]			
		BSC[3:0]				WDT[3:0]			
	IPR09	MTU0(TGI0A to TGI0D)[3:0]				MTU0(TCI0V, TGI0E, TGI0F)[3:0]			
		MTU1(TGI1A, TGI1B)[3:0]				MTU1(TCI1V, TCI1U)[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
INTC	IPR10	MTU2(TGI2A, TGI2B)[3:0]				MTU2(TCI2V, TCI2U)[3:0]			
		MTU3(TGI3A to TGI3D)[3:0]				MTU3(TCI3V)[3:0]			
	IPR11	MTU4(TGI4A to TGI4D)[3:0]				MTU4(TCI4V)[3:0]			
		MTU5(TGI5U, TGI5V, TGI5W)[3:0]				POE2(OEI1, OEI2)[3:0]			
	IPR12	MTU3S(TGI3A to TGI3D)[3:0]				MTU3S(TCI3V)[3:0]			
		MTU4S(TGI4A to TGI4D)[3:0]				MTU4S(TCI4V) [3:0]			
	IPR13	MTU5S(TGI5U, TGI5V, TGI5W)[3:0]				POE2(OEI3) [3:0]			
		IIC3[3:0]				—	—	—	—
	IPR14	SCIF0[3:0]				SCIF1[3:0]			
		SCIF2[3:0]				SCIF3[3:0]			
	IPR15	WAVEIF[3:0]				—	—	—	—
		—	—	—	—	—	—	—	—
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[1:0]	
		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[1:0]	
		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[1:0]	
		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[1:0]	
		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	—	—	—	—

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	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY[1:0]		DMAIW[2]
		DMAIW[1:0]		DMAIWA	—	—	HIZCKIO	HIZMEM	HIZCNT
	CS0BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS1BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS2BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS3BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS4BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS5BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	CS6BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS7BCR	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2]
		IWRWS[1:0]		IWRRD[2:0]			IWRRS[2:0]		
		—	TYPE[2:0]			ENDIAN	BSZ[1:0]		—
		—	—	—	—	—	—	—	—
	CS0WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS0WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1:0]		—	—	BW[1:0]	
		—	—	—	—	—	W[3:1]		
		W[0]	WM	—	—	—	—	—	—
	CS0WCR*4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	BW[1:0]	
		—	—	—	—	—	W[3:1]		
		W[0]	WM	—	—	—	—	—	—
	CS1WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS2WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3:1]		
		WR[0]	WM	—	—	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
BSC	CS2WCR*3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	A2CL[1]
		A2CL[0]	—	—	—	—	—	—	—
	CS3WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3:1]		
		WR[0]	WM	—	—	—	—	—	—
	CS3WCR*3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	WTRP[1:0]		—	WTRCD[1:0]		—	A3CL[1]
		A3CL[0]	—	—	TRWL[1:0]		—	WTRC[1:0]	
	CS4WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS4WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1:0]		—	—	BW[1:0]	
		—	—	—	SW[1:0]		W[3:1]		
		W[0]	WM	—	—	—	—	HW[1:0]	
	CS5WCR*1	—	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW/BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	CS6WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	

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	CS7WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2:0]		
		—	—	—	SW[1:0]		WR[3:1]		
		WR[0]	WM	—	—	—	—	HW[1:0]	
	SDCR	—	—	—	—	—	—	—	—
		—	—	—	A2ROW[1:0]		—	A2COL[1:0]	
		—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV
		—	—	—	A3ROW[1:0]		—	A3COL[1:0]	
	RTCSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMF	CMIE	CKS[2:0]			RRC[2:0]		
	RTCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	RTCOR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
DMAC	SAR_0								
	DAR_0								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	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								
	DAR1								

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_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								

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_2	—	—	—	—	—	—	—	—
	CHCR_2	TC	—	—	RLD	—	—	—	—
		DO	—	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_2								
	RDAR_2								
	RDMATCR_2	—	—	—	—	—	—	—	—
	SAR_3								
	DAR_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	DMATCR_3	—	—	—	—	—	—	—	—
	CHCR_3	TC	—	—	RLD	—	—	—	—
		DO	—	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_3								
	RDAR_3								
	RDMATCR_3	—	—	—	—	—	—	—	—
	SAR_4								
	DAR_4								

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_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	—	—	—	—	—	—	—	—
	SAR_5								
	DAR_5								

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
DMAC	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								
	DAR_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	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	CH1 MID[5:0]						CH1 RID[1:0]	
		CH0 MID[5:0]						CH0 RID[1:0]	
	DMARS1	CH3 MID[5:0]						CH3 RID[1:0]	
		CH2 MID[5:0]						CH2 RID[1:0]	
	DMARS2	CH5 MID[5:0]						CH5 RID[1:0]	
		CH4 MID[5:0]						CH4 RID[1:0]	
	DMARS3	CH7 MID[5:0]						CH7 RID[1:0]	
		CH6 MID[5:0]						CH6 RID[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								
	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								

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	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]			
	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								

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	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								
	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								

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	TCNTS								
	TCBR								
	TITCR	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
	TITCNT	—	3ACNT[2:0]			—	4VCNT[2:0]		
	TBTER	—	—	—	—	—	—	BTE[1:0]	
	TDER	—	—	—	—	—	—	—	TDER
	TSYCR	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
	TWCR	CCE	—	—	—	—	—	—	WRE
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
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								

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	TGRB_4S								
	TGRC_4S								
	TGRD_4S								
	TBTM_4S	—	—	—	—	—	—	TTSB	TTSA
	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								
	TCNTCMPCLRS	—	—	—	—	—	CMPCLR 5U	CMPCLR 5V	CMPCLR 5W
	TSTRS	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYRS	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TRWERS	—	—	—	—	—	—	—	RWE
	TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B

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	TOCR1S	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
	TOCR2S	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCRS	—	BDC	N	P	FB	WF	VF	UF
	TCDRS								
	TDDRS								
	TCNTSS								
	TCBRS								
	TITCRS	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]		
	TITCNTS	—	3ACNT[2:0]			—	4VCNT[2:0]		
	TBTERS	—	—	—	—	—	—	BTE[1:0]	
	TDERS	—	—	—	—	—	—	—	TDER
	TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
	TWCRS	CCE	—	—	—	—	—	SCC	WRE
	TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
POE2	ICSR1	POE3F	—	POE1F	POE0F	—	—	—	PIE1
		POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[1:0]	
	OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1
		—	—	—	—	—	—	—	—
	ICSR2	POE7F	—	—	POE4F	—	—	—	PIE2
		POE7M[1:0]		—	—	—	—	POE4M[1:0]	
	OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2
		—	—	—	—	—	—	—	—
	ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3
		—	—	—	—	—	—	POE8M[1:0]	
	SPOER	—	—	—	—	—	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ

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	POECR1	—	—	—	—	MTU2 PA25ZE	MTU2 PA24ZE	MTU2 PA23ZE	MTU2 PA22ZE
	POECR2	—	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	—	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE
		—	—	—	—	—	—	—	—
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	—	—	—	—	—
SCIF	SCSMR_0	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR_0								
	SCSCR_0	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_0	—	—	—	—	—	—	—	—
	SCFSR_0	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_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	SCFCR_0	—	—	—	—	—	—	—	—
		RTRG[1:0]		TTRG[1:0]		—	TFRST	RFRST	LOOP
	SCFDR_0	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_0	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSMR_1	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR_1								
	SCSCR_1	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1:0]	
	SCFTDR_1	—	—	—	—	—	—	—	—
	SCFSR_1	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_1								
	SCFCR_1	—	—	—	—	—	—	—	—
		RTRG[1:0]		TTRG[1:0]		—	TFRST	RFRST	LOOP
	SCFDR_1	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_1	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSEMR_1	ABCS	—	—	—	—	—	—	—
	SCSMR_2	—	—	—	—	—	—	—	—
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS[1:0]	
	SCBRR_2								
	SCSCR_2	—	—	—	—	—	—	—	—
		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_2	—	—	—	—	—	—	—	—
	SCFSR_2	PER[3:0]				FER[3:0]			
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_2	—	—	—	—	—	—	—	—
	SCFCR_2	—	—	—	—	—	—	—	—
		RTRG[1:0]		TTRG[1:0]		—	TFRST	RFRST	LOOP
	SCFDR_2	—	—	—	T[4:0]				
		—	—	—	R[4:0]				
	SCSPTR_2	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCSEMR_2	ABCS	—	—	—	—	—	—	—
	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]	
	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

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
IIC3	ICCR1	ICE	RCVD	MST	TRS	CKS[3:0]			
	ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
	ICMR	MLS	WAIT	—	—	BCWP	BC[2:0]		
	ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR	SVA[6:0]							FS
	ICDRT								
	ICDRR								
	NF2CYC	—	—	—	—	—	—	—	NF2CYC
ADC	ADCR	ADST	ADCS	ACE	ADIE	—	—	TRGE	EXTRG
	ADSR	—	—	—	—	—	—	—	ADF
	ADSTRGR	—	STR6	STR5	STR4	STR3	STR2	STR1	STR0
	ADANSR	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
	ADDR0	—	—	—	—	ADD0[11:8]			
		ADD0[7:0]							
	ADDR1	—	—	—	—	ADD1[11:8]			
		ADD1[7:0]							
	ADDR2	—	—	—	—	ADD2[11:8]			
		ADD2[7:0]							
	ADDR3	—	—	—	—	ADD3[11:8]			
		ADD3[7:0]							
	ADDR4	—	—	—	—	ADD4[11:8]			
		ADD4[7:0]							
	ADDR5	—	—	—	—	ADD5[11:8]			
		ADD5[7:0]							
	ADDR6	—	—	—	—	ADD6[11:8]			
		ADD6[7:0]							
	ADDR7	—	—	—	—	ADD7[11:8]			
		ADD7[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
DAC	DADR0								
	DADR1								
	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—
PFC	PAIORH	—	—	—	—	—	—	PA25IOR	PA24IOR
		PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR
	PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR
		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR
	PACRH3	—	—	—	—	—	—	—	—
		—	PA25MD[2:0]			—	PA24MD[2:0]		
	PACRH2	—	PA23MD[2:0]			—	PA22MD[2:0]		
		—	PA21MD[2:0]			—	PA20MD[2:0]		
	PACRH1	—	PA19MD[2:0]			—	PA18MD[2:0]		
		—	PA17MD[2:0]			—	PA16MD[2:0]		
	PACRL4	—	PA15MD[2:0]			—	PA14MD[2:0]		
		—	PA13MD[2:0]			—	PA12MD[2:0]		
	PACRL3	—	PA11MD[2:0]			—	PA10MD[2:0]		
		—	PA9MD[2:0]			—	PA8MD[2:0]		
	PACRL2	—	PA7MD[2:0]			—	PA6MD[2:0]		
		—	PA5MD[2:0]			—	PA4MD[2:0]		
	PACRL1	—	PA3MD[2:0]			—	PA2MD[2:0]		
		—	PA1MD[2:0]			—	PA0MD[2:0]		
	PBIORH	—	PB30IOR	PB29IOR	PB28IOR	PB27IOR	PB26IOR	PB25IOR	PB24IOR
		PB23IOR	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR	PB16IOR
	PBIORL	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR
	PBCRH4	—	—	—	—	—	PB30MD[2:0]		
		—	PB29MD[2:0]			—	PB28MD[2:0]		
	PBCRH3	—	PB27MD[2:0]			—	PB26MD[2:0]		
		—	PB25MD[2:0]			—	PB24MD[2:0]		

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
PFC	PBCRH2	—	PB23MD[2:0]			—	PB22MD[2:0]		
		—	PB21MD[2:0]			—	PB20MD[2:0]		
	PBCRH1	—	PB19MD[2:0]			—	PB18MD[2:0]		
		—	PB17MD[2:0]			—	PB16MD[2:0]		
	PBCRL4	—	PB15MD[2:0]			—	PB14MD[2:0]		
		—	PB13MD[2:0]			—	PB12MD[2:0]		
	PBCRL3	—	PB11MD[2:0]			—	PB10MD[2:0]		
		—	PB9MD[2:0]			—	PB8MD[2:0]		
	PBCRL2	—	PB7MD[2:0]			—	PB6MD[2:0]		
		—	PB5MD[2:0]			—	PB4MD[2:0]		
	PBCRL1	—	PB3MD[2:0]			—	PB2MD[2:0]		
		—	PB1MD[2:0]			—	PB0MD[2:0]		
	PDIOR	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	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]		
	PFCRL1	—	—	—	—	—	—	—	—
		—	PF1MD[2:0]			—	PF0MD[2:0]		
	IFCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	IRQMD[1:0]	
	WAVECR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	WVRMD[2:0]		
	WAVECR1	—	WVTMD[2:0]			—	WVSMD[2:0]		
		—	—	—	—	—	—	—	—

Module Name	Register Abbreviation	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
I/O port	PADRH	—	—	—	—	—	—	PA25DR	PA24DR
		PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR
	PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PAPRH	—	—	—	—	—	—	PA25PR	PA24PR
		PA23PR	PA22PR	PA21PR	PA20PR	PA19PR	PA18PR	PA17PR	PA16PR
	PAPRL	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR
		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
	PBDRH	—	PB30DR	PB29DR	PB28DR	PB27DR	PB26DR	PB25DR	PB24DR
		PB23DR	PB22DR	PB21DR	PB20DR	PB19DR	PB18DR	PB17DR	PB16DR
	PBDRL	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
	PBPRH	—	PB30PR	PB29PR	PB28PR	PB27PR	PB26PR	PB25PR	PB24PR
		PB23PR	PB22PR	PB21PR	PB20PR	PB19PR	PB18PR	PB17PR	PB16PR
	PBPRL	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	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
	PFDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PF1DR	PF0DR
FLASH	FCCS	FWE	MAT	—	FLER	—	—	—	SCO
	FPCS	—	—	—	—	—	—	—	PPVS
	FECS	—	—	—	—	—	—	—	EPVB
	FKEY	K[7:0]							
	FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
	FTDAR	TDER	TDA[6:0]						

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
Power-down mode	STBCR	STBY	—	—	—	—	—	—	—
	STBCR2	MSTP10	MSTP9	MSTP8	—	—	—	—	—
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	STBCR3	HIZ	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	MSTP31	MSTP30
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	—	MSTP42	—	—
H-UDI	SDIR	TI[7:0]							
		—	—	—	—	—	—	—	—

Notes: 1. When normal memory, SRAM with byte selection, or MPX-I/O is the memory type
2. When burst ROM (clocked asynchronous) is the memory type
3. When SDRAM is the memory type
4. When burst ROM (clocked synchronous) is the memory type

26.3 Register States in Each Operating Mode

Module Name	Register Abbreviation	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
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
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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
UBC	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
BSC	CMNCR	Initialized	Retained	Retained	—	Retained
	CS0BCR	Initialized	Retained	Retained	—	Retained
	CS1BCR	Initialized	Retained	Retained	—	Retained
	CS2BCR	Initialized	Retained	Retained	—	Retained
	CS3BCR	Initialized	Retained	Retained	—	Retained
	CS4BCR	Initialized	Retained	Retained	—	Retained
	CS5BCR	Initialized	Retained	Retained	—	Retained
	CS6BCR	Initialized	Retained	Retained	—	Retained
	CS7BCR	Initialized	Retained	Retained	—	Retained
	CS0WCR	Initialized	Retained	Retained	—	Retained
	CS1WCR	Initialized	Retained	Retained	—	Retained
	CS2WCR	Initialized	Retained	Retained	—	Retained
	CS3WCR	Initialized	Retained	Retained	—	Retained
	CS4WCR	Initialized	Retained	Retained	—	Retained
	CS5WCR	Initialized	Retained	Retained	—	Retained
	CS6WCR	Initialized	Retained	Retained	—	Retained
	CS7WCR	Initialized	Retained	Retained	—	Retained
	SDCR	Initialized	Retained	Retained	—	Retained
	RTCSR	Initialized	Retained (Flag processing continued)	Retained	—	Retained (Flag processing continued)
	RTCNT	Initialized	Retained (Count-up continued)	Retained	—	Retained (Count-up continued)
	RTCOR	Initialized	Retained	Retained	—	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
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
	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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
DMAC	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
	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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
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
	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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	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
	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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	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
	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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU	TDER	Initialized	Retained	Retained	Initialized	Retained
	TSYCR	Initialized	Retained	Retained	Initialized	Retained
	TWCR	Initialized	Retained	Retained	Initialized	Retained
	TOLBR	Initialized	Retained	Retained	Initialized	Retained
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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TADCOBRA_4S	Initialized	Retained	Retained	Initialized	Retained
	TADCOBRB_4S	Initialized	Retained	Retained	Initialized	Retained
	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

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TITCNTS	Initialized	Retained	Retained	Initialized	Retained
	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	Retained
	OCSR1	Initialized	Retained	Retained	Retained	Retained
	ICSR2	Initialized	Retained	Retained	Retained	Retained
	OCSR2	Initialized	Retained	Retained	Retained	Retained
	ICSR3	Initialized	Retained	Retained	Retained	Retained
	SPOER	Initialized	Retained	Retained	Retained	Retained
	POECR1	Initialized	Retained	Retained	Retained	Retained
	POECR2	Initialized	Retained	Retained	Retained	Retained
CMT	CMSTR	Initialized	Retained	Initialized	Retained	Initialized
	CMCSR_0	Initialized	Retained	Initialized	Retained	Initialized
	CMCNT_0	Initialized	Retained	Initialized	Retained	Initialized
	CMCOR_0	Initialized	Retained	Initialized	Retained	Initialized
	CMCSR_1	Initialized	Retained	Initialized	Retained	Initialized
	CMCNT_1	Initialized	Retained	Initialized	Retained	Initialized
	CMCOR_1	Initialized	Retained	Initialized	Retained	Initialized
WDT	WTCSR	Initialized	Retained	Retained	—	Retained
	WTCNT	Initialized	Retained	Retained	—	Retained
	WRCSR	Initialized*[1]	Retained	Retained	—	Retained
SCIF	SCSMR_0	Initialized	Retained	Retained	Retained	Retained
	SCBRR_0	Initialized	Retained	Retained	Retained	Retained
	SCSCR_0	Initialized	Retained	Retained	Retained	Retained
	SCFTDR_0	Undefined	Retained	Retained	Retained	Retained
	SCFSR_0	Initialized	Retained	Retained	Retained	Retained
	SCFRDR_0	Undefined	Retained	Retained	Retained	Retained
	SCFCR_0	Initialized	Retained	Retained	Retained	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
SCIF	SCFDR_0	Initialized	Retained	Retained	Retained	Retained
	SCSPTR_0	Initialized	Retained	Retained	Retained	Retained
	SCLSR_0	Initialized	Retained	Retained	Retained	Retained
	SCSMR_1	Initialized	Retained	Retained	Retained	Retained
	SCBRR_1	Initialized	Retained	Retained	Retained	Retained
	SCSCR_1	Initialized	Retained	Retained	Retained	Retained
	SCFTDR_1	Undefined	Retained	Retained	Retained	Retained
	SCFSR_1	Initialized	Retained	Retained	Retained	Retained
	SCFRDR_1	Undefined	Retained	Retained	Retained	Retained
	SCFCR_1	Initialized	Retained	Retained	Retained	Retained
	SCFDR_1	Initialized	Retained	Retained	Retained	Retained
	SCSPTR_1	Initialized	Retained	Retained	Retained	Retained
	SCLSR_1	Initialized	Retained	Retained	Retained	Retained
	SCSEMR_1	Initialized	Retained	Retained	Retained	Retained
	SCSMR_2	Initialized	Retained	Retained	Retained	Retained
	SCBRR_2	Initialized	Retained	Retained	Retained	Retained
	SCSCR_2	Initialized	Retained	Retained	Retained	Retained
	SCFTDR_2	Undefined	Retained	Retained	Retained	Retained
	SCFSR_2	Initialized	Retained	Retained	Retained	Retained
	SCFRDR_2	Undefined	Retained	Retained	Retained	Retained
	SCFCR_2	Initialized	Retained	Retained	Retained	Retained
	SCFDR_2	Initialized	Retained	Retained	Retained	Retained
	SCSPTR_2	Initialized	Retained	Retained	Retained	Retained
	SCLSR_2	Initialized	Retained	Retained	Retained	Retained
	SCSEMR_2	Initialized	Retained	Retained	Retained	Retained
	SCSMR_3	Initialized	Retained	Retained	Retained	Retained
	SCBRR_3	Initialized	Retained	Retained	Retained	Retained
	SCSCR_3	Initialized	Retained	Retained	Retained	Retained
	SCFTDR_3	Undefined	Retained	Retained	Retained	Retained
	SCFSR_3	Initialized	Retained	Retained	Retained	Retained
	SCFRDR_3	Undefined	Retained	Retained	Retained	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
SCIF	SCFCR_3	Initialized	Retained	Retained	Retained	Retained
	SCFDR_3	Initialized	Retained	Retained	Retained	Retained
	SCSPTR_3	Initialized	Retained	Retained	Retained	Retained
	SCLSR_3	Initialized	Retained	Retained	Retained	Retained
IIC3	ICCR1	Initialized	Retained	Retained	Retained	Retained
	ICCR2	Initialized	Retained	Retained	Retained	Retained
	ICMR	Initialized	Retained	Retained/ Initialized (bc3-0)	Retained/ Initialized (bc3-0)	Retained
	ICIER	Initialized	Retained	Retained	Retained	Retained
	ICSR	Initialized	Retained	Retained	Retained	Retained
	SAR	Initialized	Retained	Retained	Retained	Retained
	ICDRT	Initialized	Retained	Retained	Retained	Retained
	ICDRR	Initialized	Retained	Retained	Retained	Retained
	NF2CYC	Initialized	Retained	Retained	Retained	Retained
ADC	ADCR	Initialized	Retained	Initialized	Retained	Retained
	ADSR	Initialized	Retained	Initialized	Retained	Retained
	ADSTRGR	Initialized	Retained	Initialized	Retained	Retained
	ADANSR	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
	ADDR4	Initialized	Retained	Initialized	Retained	Retained
	ADDR5	Initialized	Retained	Initialized	Retained	Retained
	ADDR6	Initialized	Retained	Initialized	Retained	Retained
	ADDR7	Initialized	Retained	Initialized	Retained	Retained
DAC	DADR0	Initialized	Retained	Retained	Initialized	Retained
	DADR1	Initialized	Retained	Retained	Initialized	Retained
	DACR	Initialized	Retained	Retained	Initialized	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
PFC	PAIORH	Initialized	Retained	Retained	—	Retained
	PAIORL	Initialized	Retained	Retained	—	Retained
	PACRH3	Initialized	Retained	Retained	—	Retained
	PACRH2	Initialized	Retained	Retained	—	Retained
	PACRH1	Initialized	Retained	Retained	—	Retained
	PACRL4	Initialized	Retained	Retained	—	Retained
	PACRL3	Initialized	Retained	Retained	—	Retained
	PACRL2	Initialized	Retained	Retained	—	Retained
	PACRL1	Initialized	Retained	Retained	—	Retained
	PBIORH	Initialized	Retained	Retained	—	Retained
	PBIORL	Initialized	Retained	Retained	—	Retained
	PBCRH4	Initialized	Retained	Retained	—	Retained
	PBCRH3	Initialized	Retained	Retained	—	Retained
	PBCRH2	Initialized	Retained	Retained	—	Retained
	PBCRH1	Initialized	Retained	Retained	—	Retained
	PBCRL4	Initialized	Retained	Retained	—	Retained
	PBCRL3	Initialized	Retained	Retained	—	Retained
	PBCRL2	Initialized	Retained	Retained	—	Retained
	PBCRL1	Initialized	Retained	Retained	—	Retained
	PDIOR	Initialized	Retained	Retained	—	Retained
	PDCRL4	Initialized	Retained	Retained	—	Retained
	PDCRL3	Initialized	Retained	Retained	—	Retained
	PDCRL2	Initialized	Retained	Retained	—	Retained
	PDCRL1	Initialized	Retained	Retained	—	Retained
	PFCRL1	Initialized	Retained	Retained	—	Retained
	IFCR	Initialized	Retained	Retained	—	Retained
	WAVECR2	Initialized	Retained	Retained	—	Retained
	WAVECR1	Initialized	Retained	Retained	—	Retained

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
I/O port	PADRH	Initialized	Retained	Retained	—	Retained
	PADRL	Initialized	Retained	Retained	—	Retained
	PAPRH	Undefined	Retained	Retained	—	Retained
	PAPRL	Undefined	Retained	Retained	—	Retained
	PBDRH	Initialized	Retained	Retained	—	Retained
	PBDRL	Initialized	Retained	Retained	—	Retained
	PBPRH	Undefined	Retained	Retained	—	Retained
	PBPRL	Undefined	Retained	Retained	—	Retained
	PDDRL	Initialized	Retained	Retained	—	Retained
	PDPRL	Undefined	Retained	Retained	—	Retained
	PFDR	Initialized	Retained	Retained	—	Retained
FLASH	FCCS	Initialized	Retained	Initialized	Initialized	Retained
	FPCS	Initialized	Retained	Initialized	Initialized	Retained
	FECS	Initialized	Retained	Initialized	Initialized	Retained
	FKEY	Initialized	Retained	Initialized	Initialized	Retained
	FMATS	Initialized	Retained	Initialized	Initialized	Retained
	FTDAR	Initialized	Retained	Initialized	Initialized	Retained
Power-down mode	STBCR	Initialized	Retained	Retained	—	Retained
	STBCR2	Initialized	Retained	Retained	—	Retained
	SYSCR1	Initialized	Retained	Retained	—	Retained
	SYSCR2	Initialized	Retained	Retained	—	Retained
	STBCR3	Initialized	Retained	Retained	—	Retained
	STBCR4	Initialized	Retained	Retained	—	Retained
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 $\overline{\text{TRST}}$ assertion or in the Test-Logic-Reset state of the TAP controller.

Section 27 Electrical Characteristics

Note: The current specifications of this section are provisional. Note that they are subject to change without notice.

27.1 Absolute Maximum Ratings

Table 27.1 lists the absolute maximum ratings.

Table 27.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage (I/O)	$V_{CC}Q$	–0.3 to 4.6	V
Power supply voltage (Internal)	V_{CC} $PLLV_{CC}$	–0.3 to 2.3	V
Input voltage (except analog input pins)	Vin	–0.3 to $V_{CC}Q$ +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to 7.0	V
Analog reference voltage	AVREF	–0.3 to AV_{CC} +0.3	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	
Operating temperature	T_{opr}	–40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

27.2 Power-on/Power-off Sequence

Power-on/power-off sequence and their recommended values are shown below.

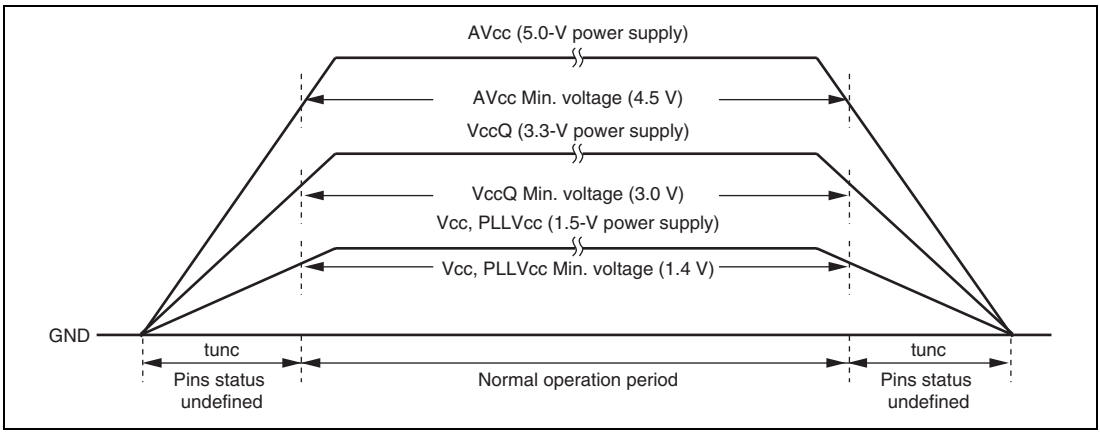


Figure 27.1 Power-on/Power-off Sequence

Table 27.2 Recommended Time for Power-on/Power-off Sequence

Item	Symbol	Maximum Allowance Value	Unit
Undefined time	tunc	100	ms

Note: VccQ ≥ Vcc = PLLVcc is recommended. Either VccQ, Vcc, or PLLVcc power supply can be turned on or off first, though, an undefined period appears until Vcc rises to the Min. voltage or after Vcc passes the Min. voltage. During these periods, pin or internal states become undefined. Design the system so that such undefined states do not cause a system malfunction. To avoid an increase in the current consumption during the undefined period at power-on, it is recommended that VccQ, Vcc, and PLLVcc be turned on simultaneously. This undefined period can be eliminated by turning on the power supplies in the order shown in figure 27.2.

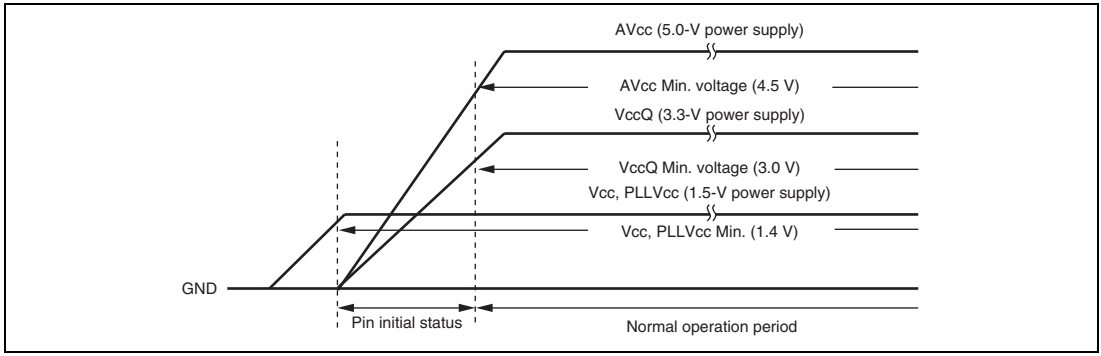


Figure 27.2 Power-On Sequence

Notes: 1. AVcc = AVREF > Vcc = PLLVcc is recommended. Either AVcc or VccQ power supply can be turned on or off first. However, note that the equation AVcc ± 0.3 V = AVREF should always be true. Using the LSI under the following conditions may result in decreased reliability or permanent damage to the LSI.

AVREF > AVcc ± 0.3 V

2. To prevent the pin and internal states from being undefined, VccQ and AVcc should be kept GND voltage level (0 V) and they should not be placed in floating state until Vcc reaches the Min. voltage. In addition, the $\overline{\text{RES}}$ pin should be input low to place power-on reset state. In this case, care must be taken for the power consumption increase caused by sink current because each pin is placed in low-impedance state until VccQ reaches the Min. voltage.

27.3 DC Characteristics

Table 27.3 lists DC characteristics.

Table 27.3 DC Characteristics (1) [Common Items]

Conditions: Ta = –40°C to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power supply voltage		$V_{CC}Q$	3.0	3.3	3.6	V	
		V_{CC} $PLLV_{CC}$	1.4	1.5	1.6	V	
Analog power supply voltage		AV_{CC}	4.5	5.0	5.5	V	
Current consumption*1	Normal operation	I_{CC}	–	200	310	mA	V_{CC} = 1.5 V Iϕ = 160 MHz Bϕ = 40 MHz Pϕ = 40 MHz
	Software standby mode	I_{stby}	–	5	60	mA	Ta = 25°C V_{CC} = 3.3 V V_{CC} = 1.5 V Bϕ = 40 MHz Pϕ = 40 MHz
		PI_{stby}	–	0.1	1	mA	
	Sleep mode	I_{sleep}	–	60	100	mA	
Input leakage current	All input pins (except PF0, PF1)	$\|I_{in}\|$	–	–	1	µA	V_{in} = 0.5 to $V_{CC}Q$ – 0.5 V
	PF0, PF1		–	–	1	µA	
Three-state leakage current	Input/output pins, all output pins (off state)	$\|I_{STI}\|$	–	–	1	µA	Vin = 0.5 to $V_{CC}Q$ – 0.5 V
Input capacitance	All pins	C_{in}	–	–	10	pF	
Analog power supply current	During A/D or D/A conversion	AI_{CC}	–	5	4	mA	Including AVREF
	Waiting for A/D or D/A conversion		–	1	3	mA	
	Standby mode			–	4	µA	

Caution: When neither the A/D converter nor the D/A converter is in use, set Vcc ≤ AVcc ≤ 5.0 V ± 0.5 V and AVss = Vss, and do not leave the AVcc, AVss, AVREF, and AVREFVss pins open.

Notes: 1. Current consumption values are when all output pins are unloaded.
2. I_{CC}, I_{sleep}, and I_{stby} represent the total currents consumed in the Vcc and PLLVcc systems.
3. PI_{stby} is the total current consumed in the VccQ power supply.

Table 27.3 DC Characteristics (2) [Except for I²C-Related Pins]

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6 V, Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD1, MD0, MD_CLK2, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, EXTAL, $\overline{ASEBRK}$, $\overline{FWE}$	V_{IH}	$V_{CC}Q - 0.5$	—	$V_{CC}Q + 0.3$	V	
	Input pins other than above (excluding Schmitt pins)		2.0	—	$V_{CC}Q + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD1, MD0, MD_CLK2, MD_CLK0, $\overline{ASEMD}$, $\overline{TRST}$, EXTAL, $\overline{ASEBRK}$, $\overline{FWE}$	V_{IL}	–0.3	—	0.5	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, TIOC3BS, TIOC3CS, TIOC3DS, TIOC4AS, TIOC4BS, TIOC4CS, TIOC4DS, TIC5US, TIC5VS, TIC5WS, $\overline{\text{POE8}}$, $\overline{\text{POE7}}$, $\overline{\text{POE4}}$, $\overline{\text{POE3}}$, $\overline{\text{POE1}}$, $\overline{\text{POE0}}$, SCK3 to SCK0, RXD3 to RXD0, IRQ7 to IRQ0	V_T^+	$V_{CC}Q - 0.5$	–	–	V	
		V_T^-	–	—	0.5	V	
		$V_T^+ - V_T^-$	0.4	–	–	V	
Output high voltage	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	V_{OH}	$V_{CC}Q - 0.8$	–	–	V	$I_{OH} = -5$ mA
	All output pins except for above pins		$V_{CC}Q - 0.5$	–	–	V	$I_{OH} = -200$ μA

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output low voltage	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	V_{OL}	–	–	0.9	V	I_{OL} = 15 mA
	All output pins except for above pins		–	–	0.4	V	I_{OL} = 2 mA

Table 27.3 DC Characteristics (3) [I²C-Related Pins*]

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	$V_{CC}Q \times 0.7$	–	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	–0.3	–	$PV_{CC} \times 0.3$	V	
Schmitt trigger input characteristics	$V_{IH} - V_{IL}$	0.4	–	–	V	
Output low voltage	V_{OL}	–	–	0.4	V	I_{OL} = 3.0 mA

Note: * The PF0/IRQ0/$\overline{\text{POE7}}$/SCL and PF1/IRQ1/$\overline{\text{POE3}}$/SDA pins (open-drain pins)

Table 27.4 Permissible Output Currents

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	I_{OL}	–	–	15	mA
	SCL, SDA				10	mA
	Output pins other than above				2	mA
Permissible output low current (total)		ΣI_{OL}	–	–	80	mA
Permissible output high current (per pin)	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	$-I_{OH}$	–	–	5	mA
	Output pins other than above		–	–	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	25	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 27.4.

27.4 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed.

Table 27.5 Maximum Operating Frequency

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

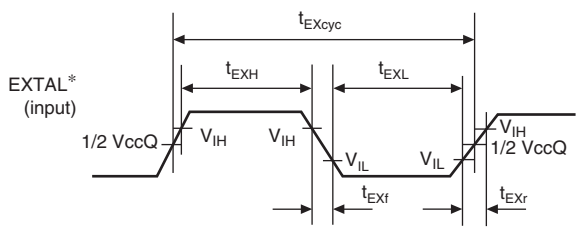
Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU (Iϕ)	f	32	–	160	MHz	
	Internal bus, external bus (Bϕ)		32	–	40		
	Peripheral module (Pϕ)		4	–	40		

27.4.1 Clock Timing

Table 27.6 Clock Timing

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	8	10	MHz	Figure 27.3
EXTAL clock input cycle time	t_{EXcyc}	100	125	ns	
EXTAL clock input pulse low width	t_{EXL}	6	–	ns	
EXTAL clock input pulse high width	t_{EXH}	6	–	ns	
EXTAL clock input rise time	t_{EXr}	–	3	ns	
EXTAL clock input fall time	t_{EXf}	–	3	ns	
CK clock output frequency	f_{OP}	16	40	MHz	Figure 27.4
CK clock output cycle time	t_{cyc}	25.0	62.5	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 27.5
Oscillation settling time on return from standby 1	t_{OSC2}	10	–	ms	Figure 27.6
Oscillation settling time on return from standby 2	t_{OSC3}	10	–	ms	Figure 27.7



Note: * When the clock is input on the EXTAL pin.

Figure 27.3 EXTAL Clock Input Timing

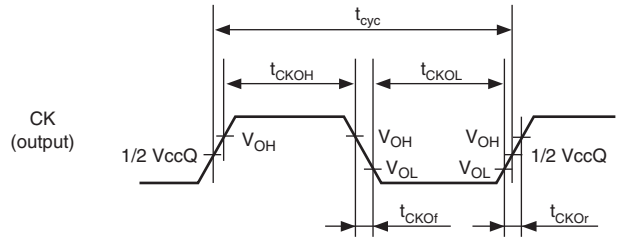
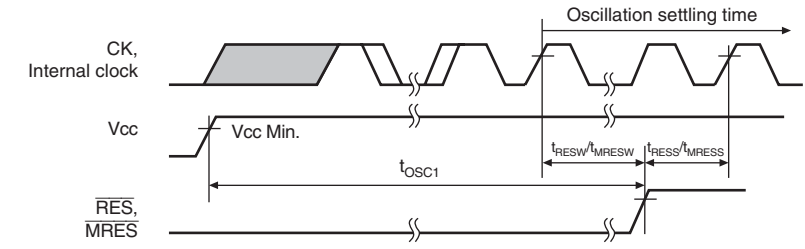
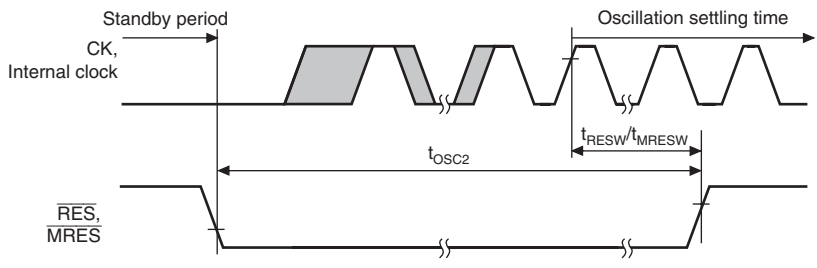


Figure 27.4 CK Clock Output Timing



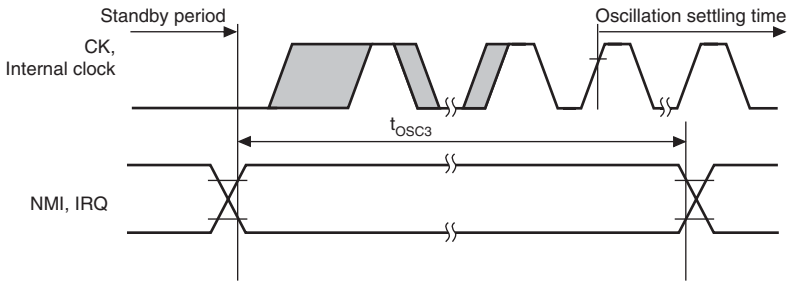
Note: Oscillation settling time when the internal oscillator is used.

Figure 27.5 Power-On Oscillation Settling Time



Note: Oscillation settling time when the internal oscillator is used.

Figure 27.6 Oscillation Settling Time on Return from Standby (Return by Reset)



Note: Oscillation settling time when the internal oscillator is used.

Figure 27.7 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

27.4.2 Control Signal Timing

Table 27.7 Control Signal Timing

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

		Bϕ = 40 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{\text{RES}}$ pulse width	t_{RESW}	20*[2]	–	t_{cyc}	Figure 27.5, Figure 27.6, Figure 27.8, Figure 27.9
$\overline{\text{RES}}$ setup time *[1]	t_{RESS}	25	–	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}	25	–	ns	
$\overline{\text{MRES}}$ hold time	t_{MRESH}	15	–	ns	
MD1, MD0 setup time	t_{MDS}	20	–	t_{cyc}	Figure 27.8
$\overline{\text{BREQ}}$ setup time	t_{BREQS}	$1/2t_{cyc}$ + 10	–	ns	Figure 27.10
$\overline{\text{BREQ}}$ hold time	t_{BREQH}	$1/2t_{cyc}$ + 4	–	ns	
NMI setup time *[1]	t_{NMIS}	15	–	ns	Figure 27.9
NMI hold time	t_{NMIH}	7	–	ns	
IRQ7 to IRQ0 setup time *[1]	t_{IRQS}	15	–	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	7	–	ns	
$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ output delay time	t_{IRQOD}	—	100	ns	Figure 27.11
$\overline{\text{BACK}}$ delay time	t_{BACKD}	—	$1/2t_{cyc}$ + 20	ns	Figure 27.10
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 IRQ7 to IRQ0 are asynchronous signals. When these setup times are observed, a change of these signals is detected at the clock rising edge. If the setup times are not observed, detection of a signal change may be delayed until the next rising edge of the clock.

2. In standby mode or when the clock multiplication ratio is changed, $t_{RESW} = t_{OSC2}$ (Min. 10 ms).

3. In standby mode, $t_{RESW} = t_{OSC2}$ (Min. 10 ms).

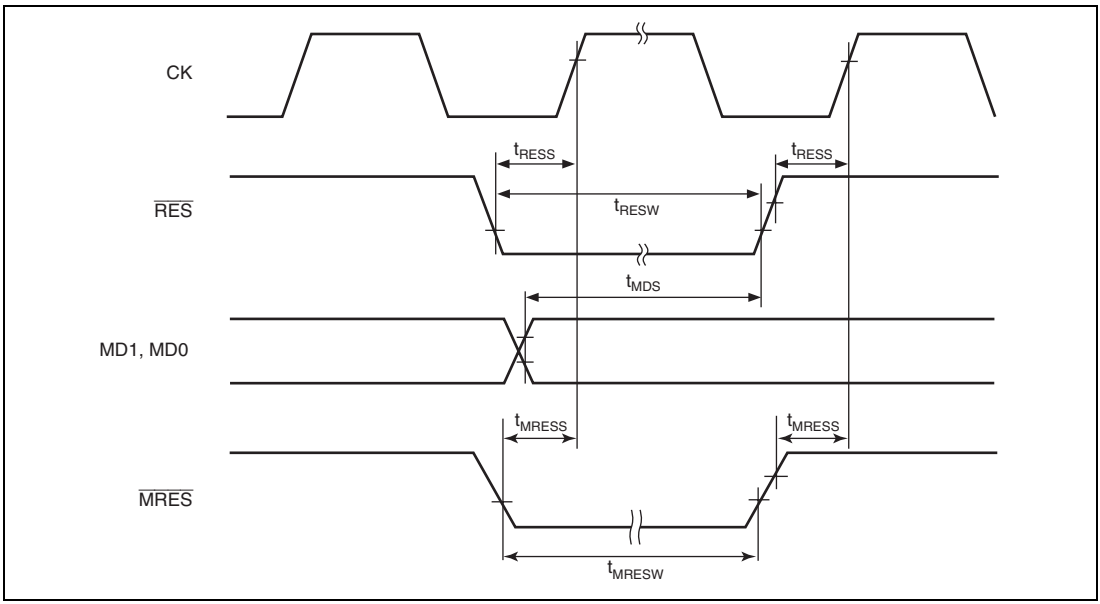


Figure 27.8 Reset Input Timing

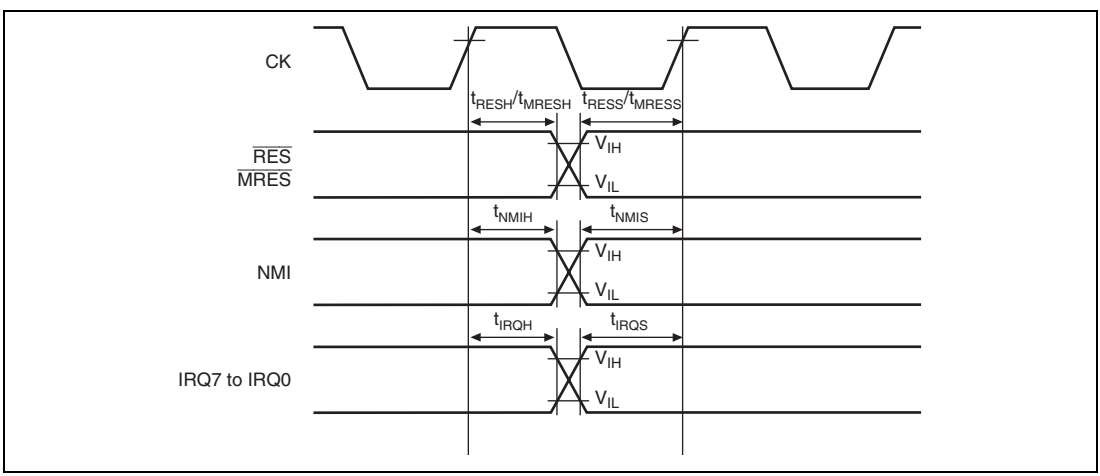


Figure 27.9 Interrupt Signal Input Timing

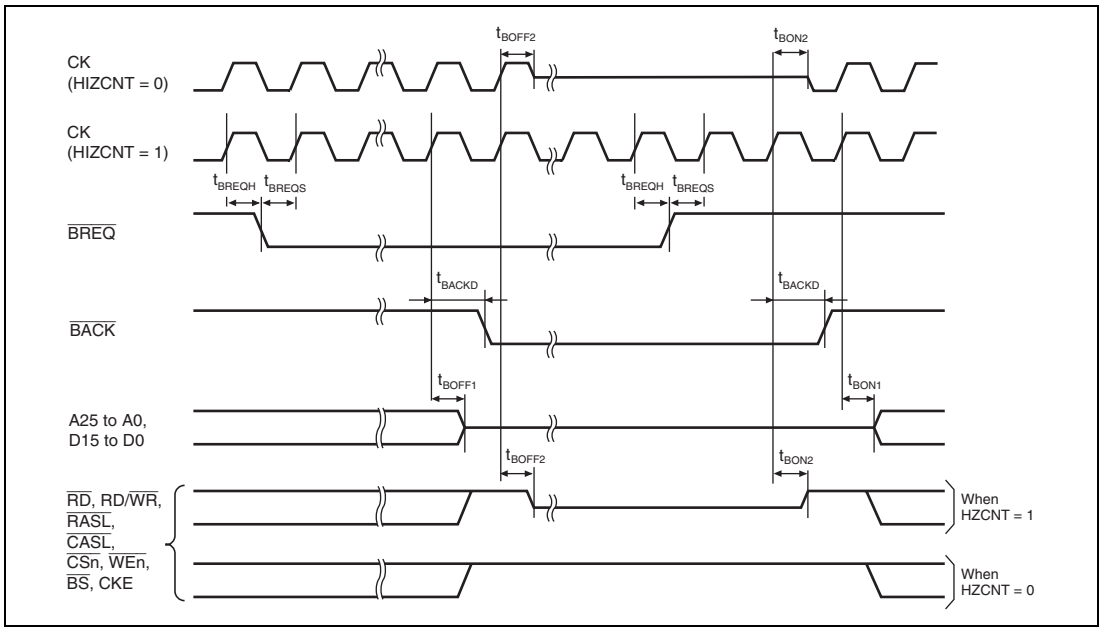


Figure 27.10 Interrupt Signal Output Timing

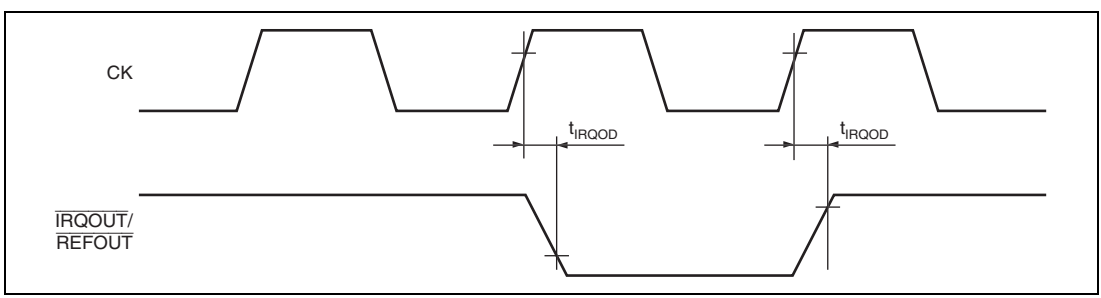


Figure 27.11 Bus Release Timing

27.4.3 Bus Timing

Table 27.8 Bus Timing

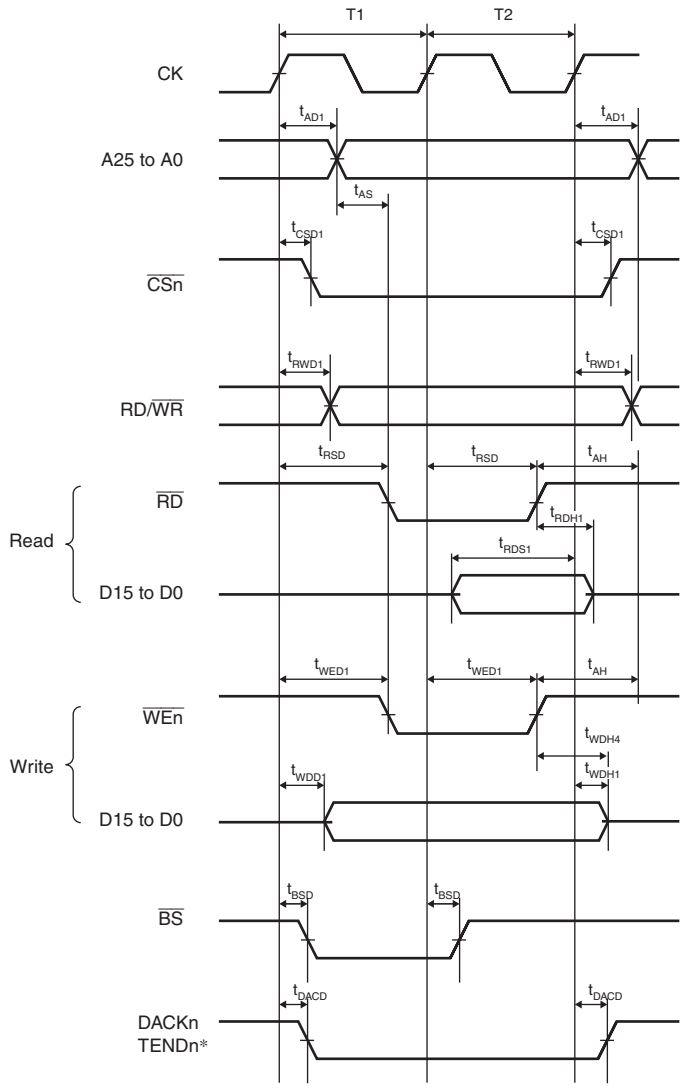
Conditions: Clock mode 6, Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V, Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

		Bϕ = 40 MHz*			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1	t_{AD1}	1	20	ns	Figures 27.12 to 27.36
Address delay time 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figure 27.19
Address delay time 3	t_{AD3}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.37, 27.38
Address setup time	t_{AS}	0	–	ns	Figures 27.12 to 27.15, 27.19
Address hold time	t_{AH}	0	–	ns	Figures 27.12 to 27.15
$\overline{BS}$ delay time	t_{BSD}	–	20	ns	Figures 27.12 to 27.33, 27.37
$\overline{CS}$ delay time 1	t_{CSD1}	1	20	ns	Figures 27.12 to 27.36
$\overline{CS}$ delay time 2	t_{CSD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.37, 27.38
Read write delay time 1	t_{RWD1}	1	20	ns	Figures 27.12 to 27.36
Read write delay time 2	t_{RWD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.37, 27.38
Read strobe delay time	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.12 to 27.19
Read data setup time 1	t_{RDS1}	$1/2t_{cyc}$ + 20	–	ns	Figures 27.12 to 27.18
Read data setup time 2	t_{RDS2}	10	–	ns	Figures 27.20 to 27.23, 27.28 to 27.30
Read data setup time 3	t_{RDS3}	$1/2t_{cyc}$ + 20	–	ns	Figure 27.19
Read data setup time 4	t_{RDS4}	$1/2t_{cyc}$ + 20	–	ns	Figure 27.37

Item	Symbol	Bϕ = 40 MHz* Min.	Max.	Unit	Figure
Read data hold time 1	t_{RDH1}	0	—	ns	Figures 27.12 to 27.18
Read data hold time 2	t_{RDH2}	2	—	ns	Figures 27.20 to 27.23, 27.28 to 27.30
Read data hold time 3	t_{RDH3}	0	—	ns	Figure 27.19
Read data hold time 4	t_{RDH4}	$1/2t_{cyc} + 5$	—	ns	Figure 27.37
Write enable delay time 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc} + 20$	ns	Figures 27.12 to 27.17
Write enable delay time 2	t_{WED2}	—	20	ns	Figure 27.18
Write data delay time 1	t_{WDD1}	—	20	ns	Figures 27.12 to 27.18
Write data delay time 2	t_{WDD2}	—	20	ns	Figures 27.24 to 27.27, 27.31 to 27.33
Write data delay time 3	t_{WDD3}	—	$1/2t_{cyc} + 20$	ns	Figure 27.37
Write data hold time 1	t_{WDH1}	1	—	ns	Figures 27.12 to 27.18
Write data hold time 2	t_{WDH2}	1	—	ns	Figures 27.24 to 27.27, 27.31 to 27.33
Write data hold time 3	t_{WDH3}	$1/2t_{cyc}$	—	ns	Figure 27.37
Write data hold time 4	t_{WDH4}	0	—	ns	Figures 27.12, 27.16
$\overline{WAIT}$ setup time	t_{WTS}	$1/2t_{cyc} + 10$	—	ns	Figures 27.13 to 27.19
$\overline{WAIT}$ hold time	t_{WTH}	$1/2t_{cyc} + 5$	—	ns	Figures 27.13 to 27.19
$\overline{RAS}$ delay time 1	t_{RASD1}	1	20	ns	Figures 27.20 to 27.31, 27.33 to 27.36
$\overline{RAS}$ delay time 2	t_{RASD2}	$1/2t_{cyc}$	$1/2t_{cyc} + 20$	ns	Figures 27.37, 27.38

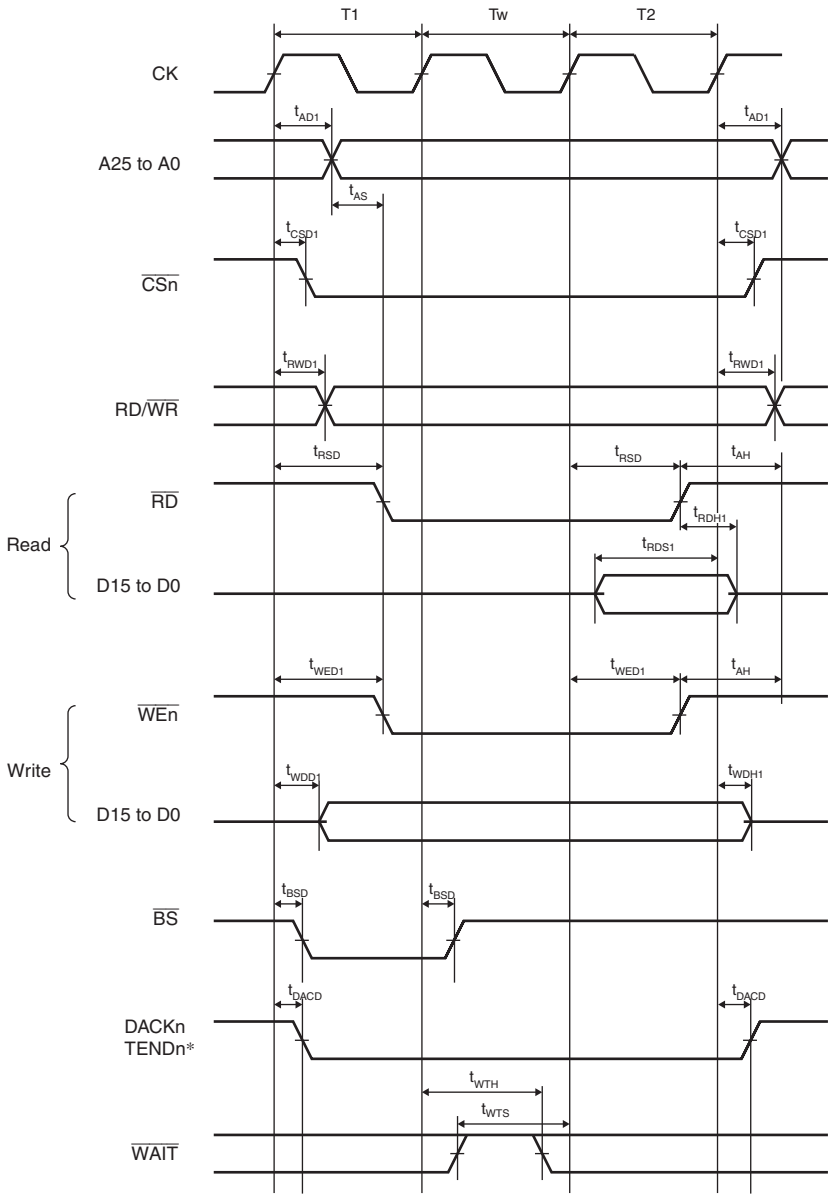
Item	Symbol	Bϕ = 40 MHz*[1] Min.	Bϕ = 40 MHz*[1] Max.	Unit	Figure
$\overline{CAS}$ delay time 1	t_{CASD1}	1	20	ns	Figures 27.20 to 27.36
$\overline{CAS}$ delay time 2	t_{CASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.37, 27.38
$\overline{DQM}$ delay time 1	t_{DQMD1}	1	20	ns	Figures 27.20 to 27.33
$\overline{DQM}$ delay time 2	t_{DQMD2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figures 27.37, 27.38
CKE delay time 1	t_{CKED1}	1	20	ns	Figure 27.35
CKE delay time 2	t_{CKED2}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figure 27.38
$\overline{AH}$ delay time	t_{AHD}	$1/2t_{cyc}$	$1/2t_{cyc}$ + 20	ns	Figure 27.16
Multiplexed address delay time	t_{MAD}	—	20	ns	Figure 27.16
Multiplexed address hold time	t_{MAH}	1	—	ns	Figure 27.16
DACK, TEND delay time	t_{DACD}	—	Refer to peripheral modules	ns	Figures 27.12 to 27.33, 27.37

Note: * The maximum value (f_{max}) of Bϕ (external bus clock) depends on the number of wait cycles and the system configuration of your board.



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.12 Basic Bus Timing for Normal Space (No Wait)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.13 Basic Bus Timing for Normal Space (One Software Wait Cycle)

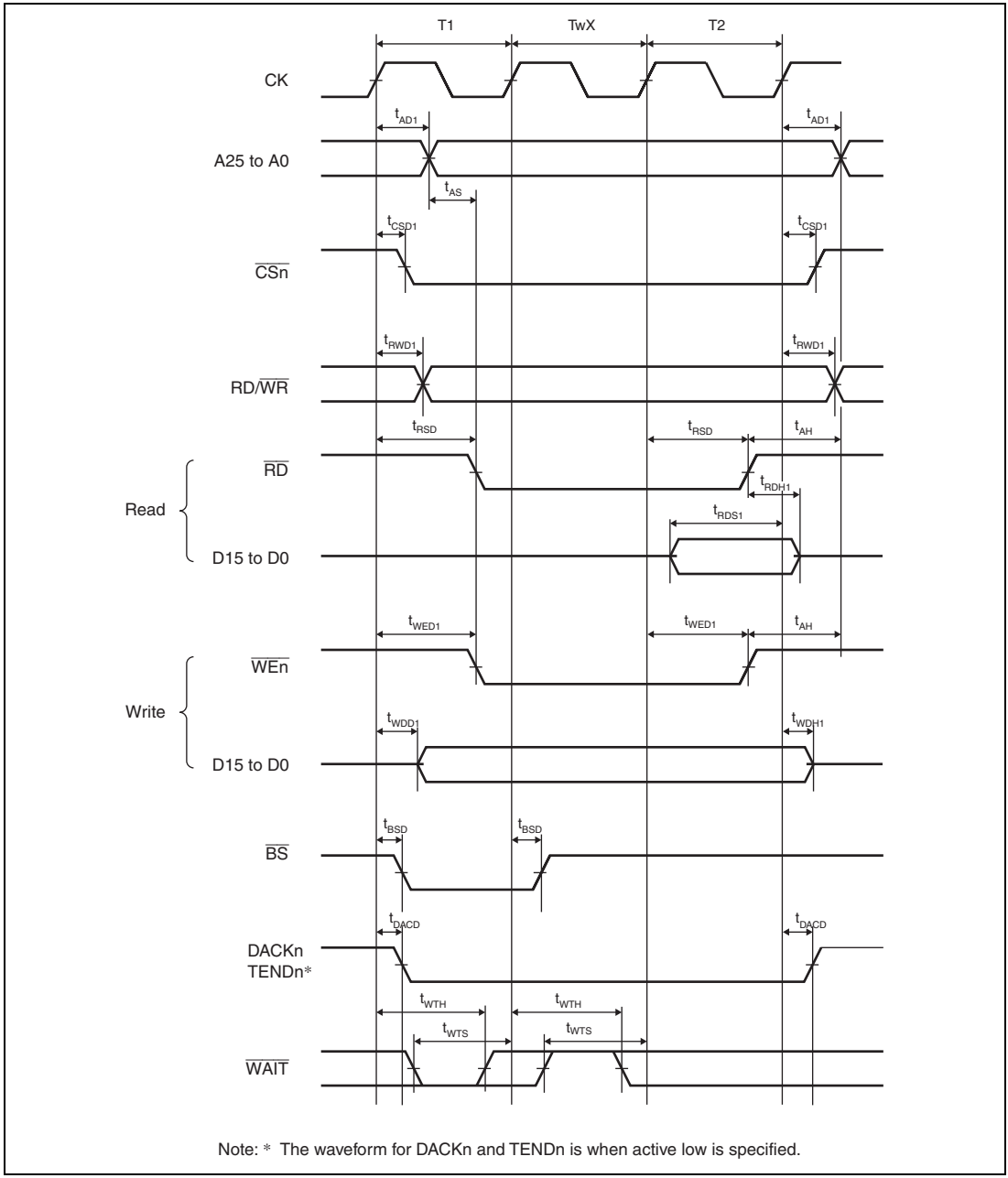
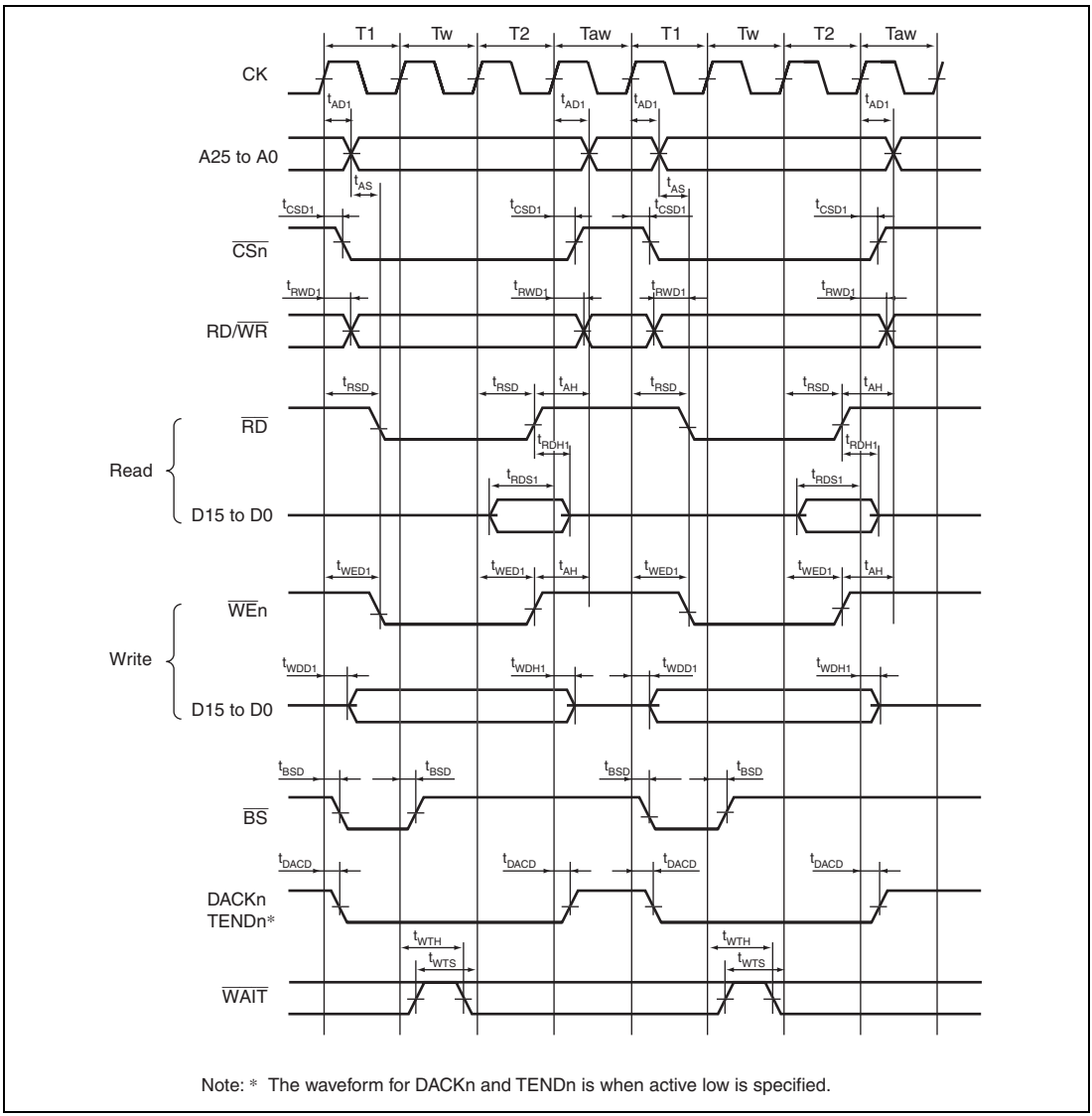
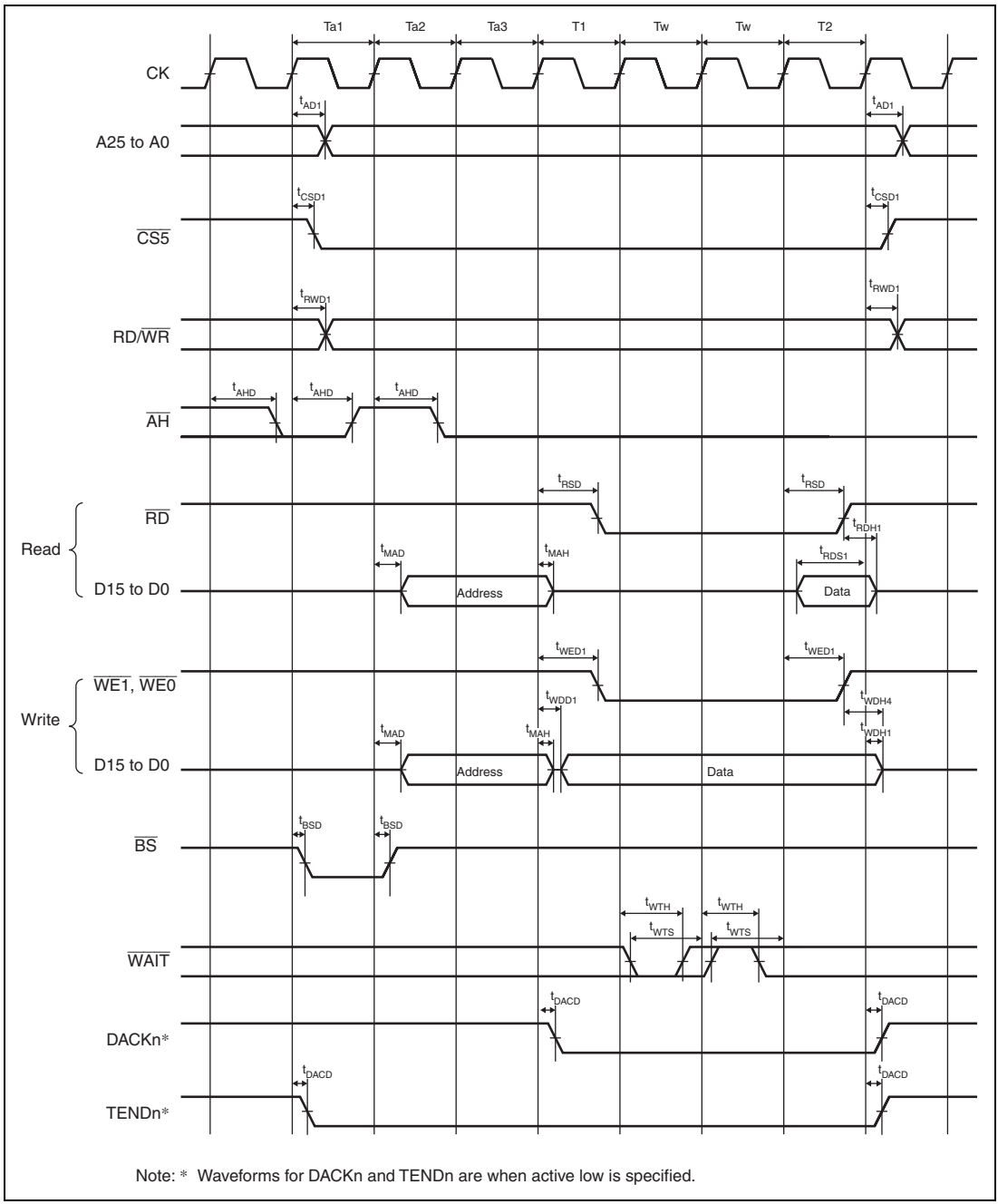


Figure 27.14 Basic Bus Timing for Normal Space (One External Wait Cycle)



Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.15 Basic Bus Timing for Normal Space (One Software Wait Cycle, External Wait Cycle Valid (WM Bit = 0), No Idle Cycle)



Note: * Waveforms for DACKn and TENDn are when active low is specified.

Figure 27.16 MPX-I/O Interface Bus Cycle
(Three Address Cycles, One Software Wait Cycle, One External Wait Cycle)

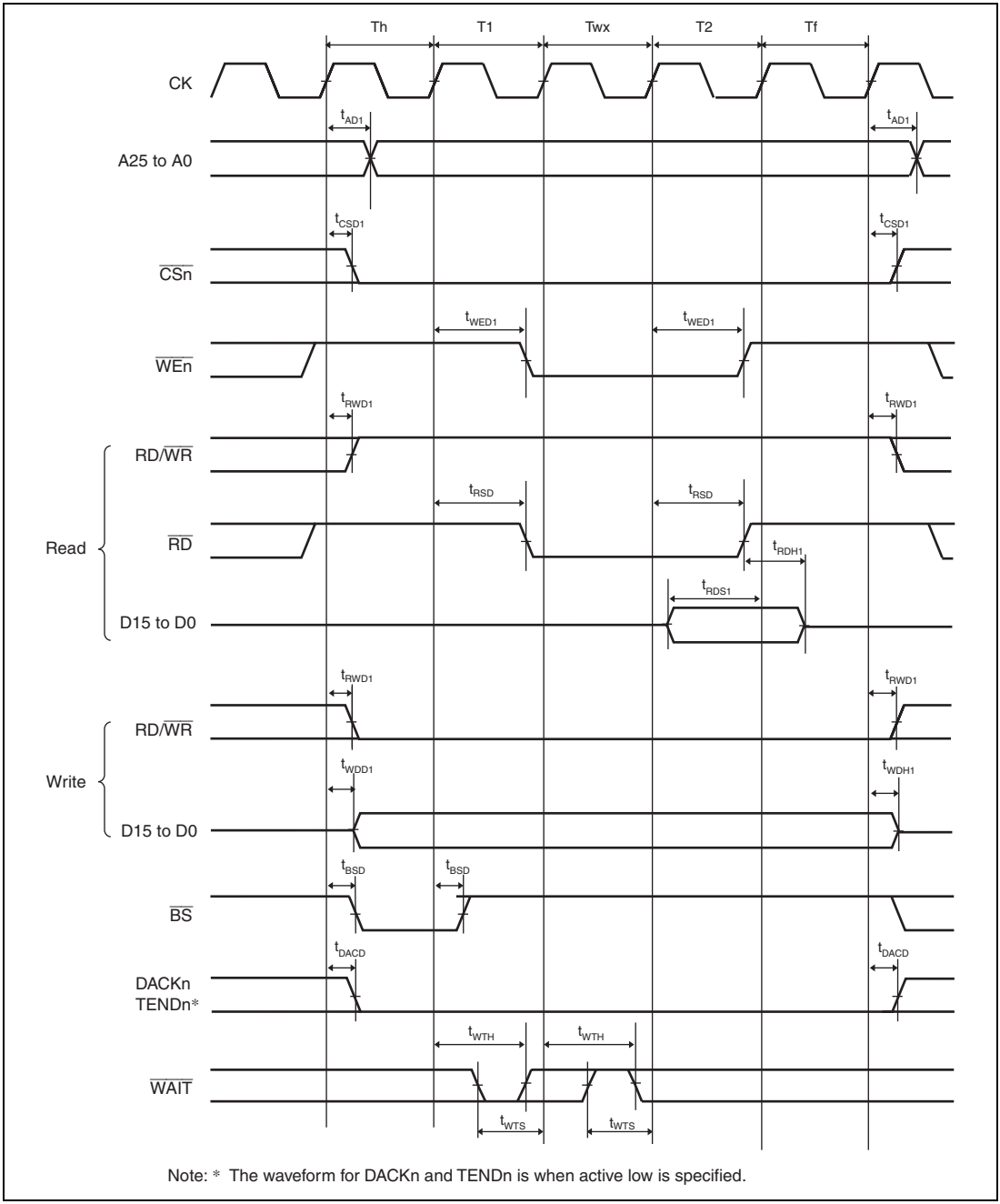


Figure 27.17 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 0 (Write Cycle UB/LB Control))

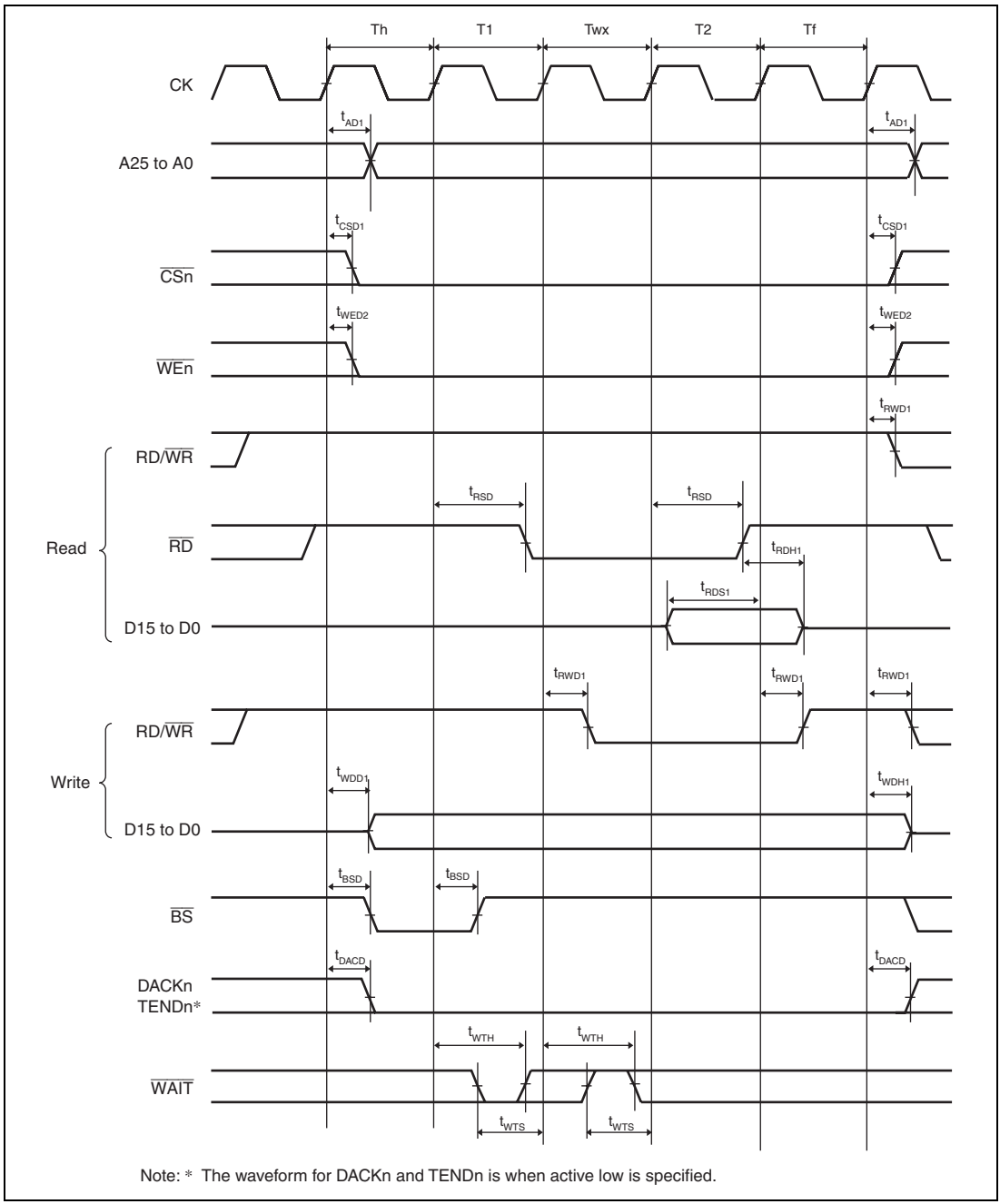
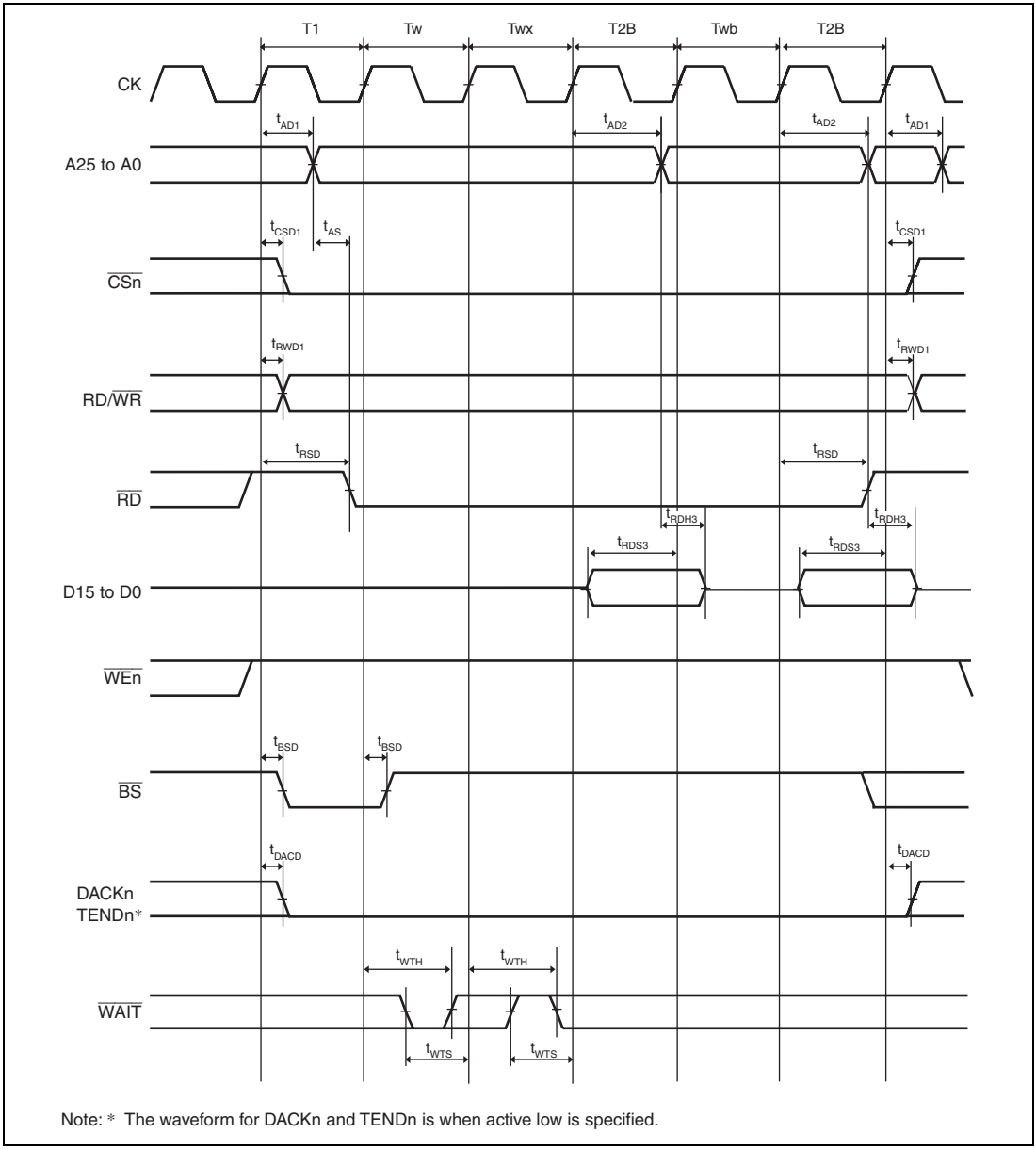
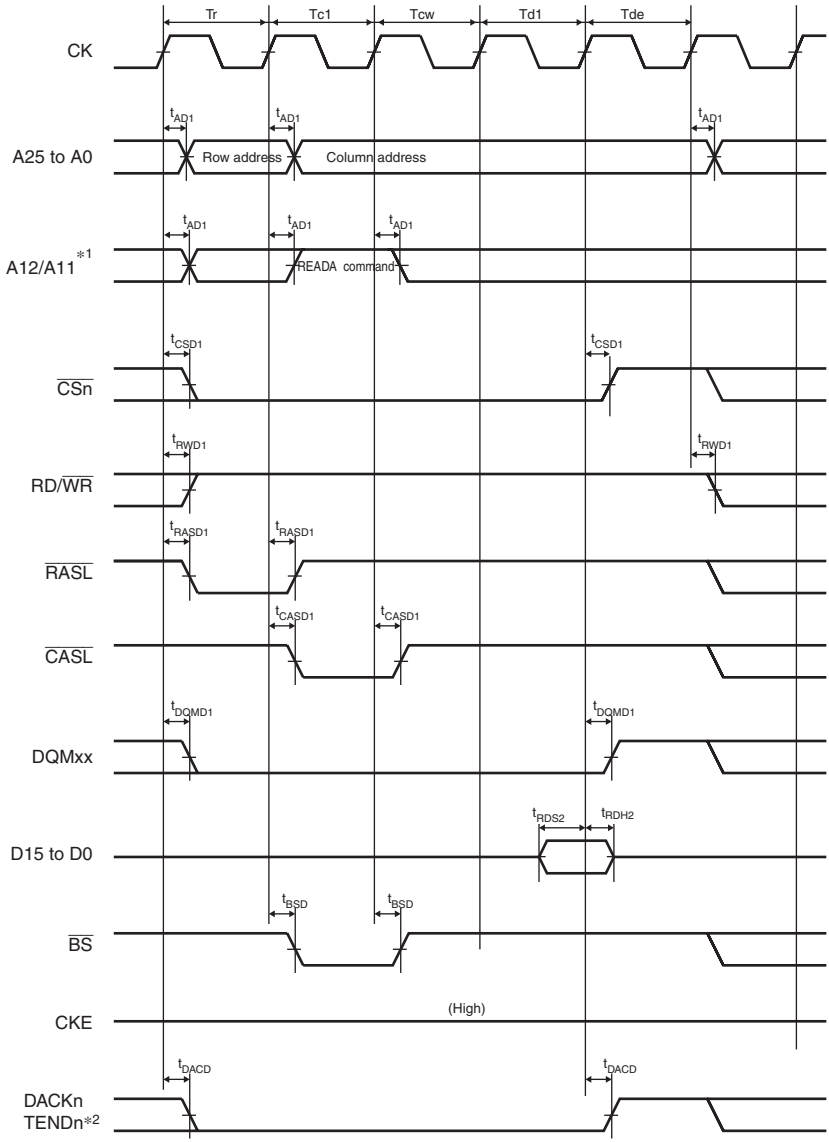


Figure 27.18 Bus Cycle of SRAM with Byte Selection (SW = 1 Cycle, HW = 1 Cycle, One Asynchronous External Wait Cycle, BAS = 1 (Write Cycle WE Control))



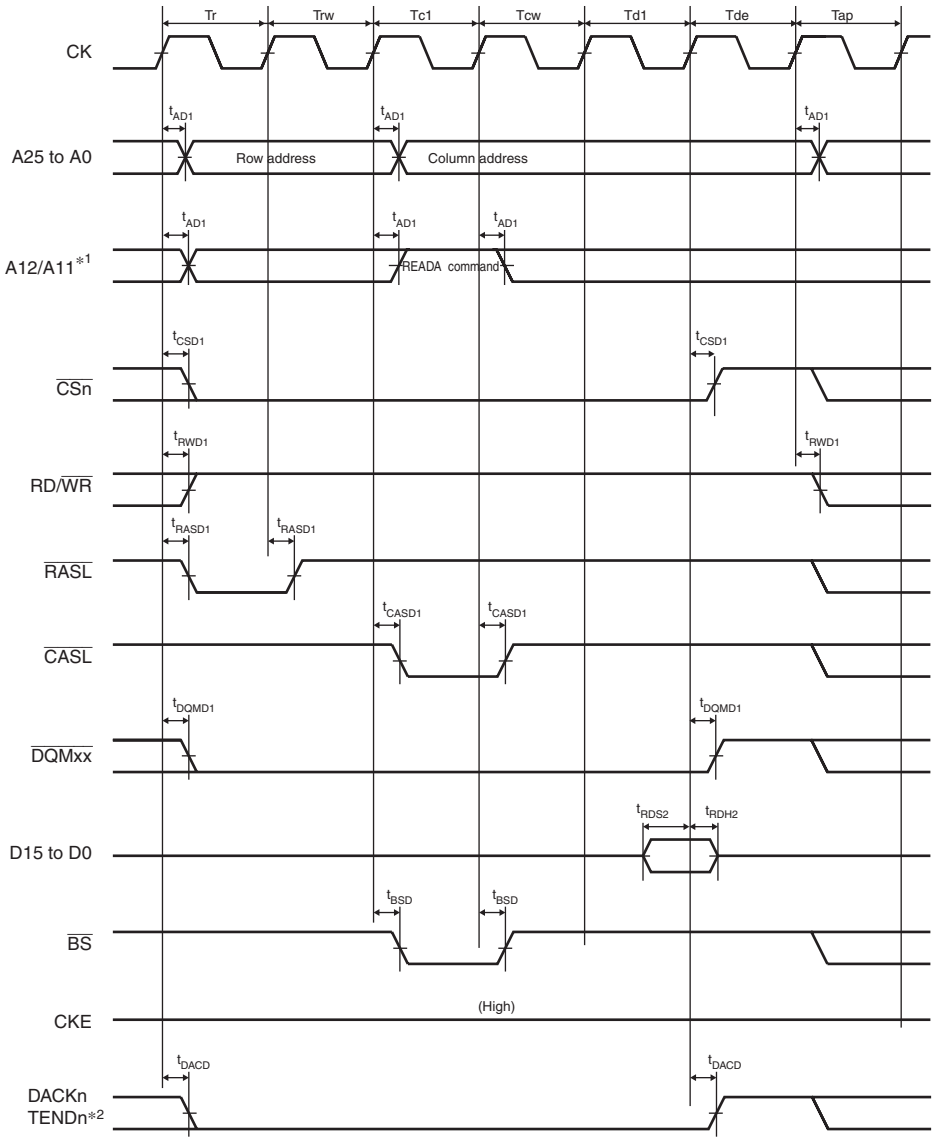
Note: * The waveform for DACKn and TENDn is when active low is specified.

Figure 27.19 Burst ROM Read Cycle
(One Software Wait Cycle, One Asynchronous External Burst Wait Cycle, Two-Cycle Burst)



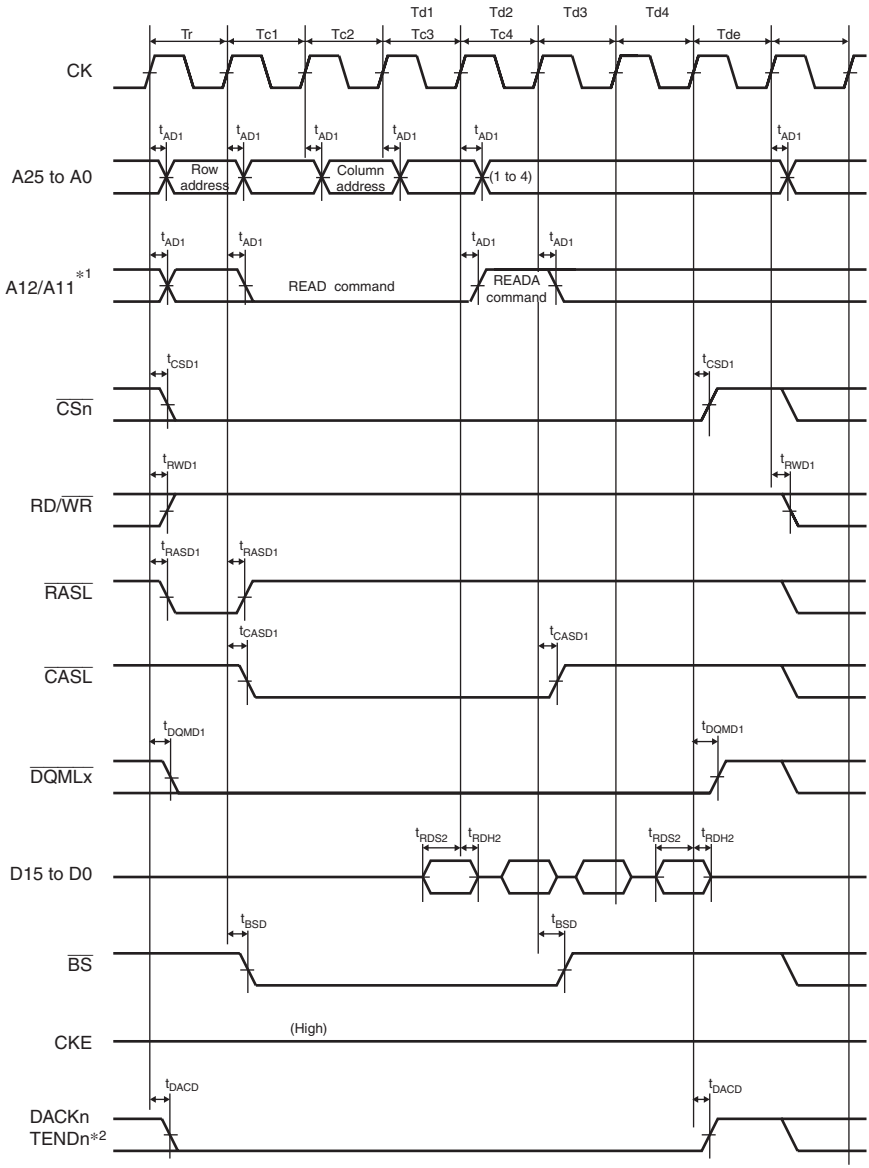
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.20 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 0 Cycle)



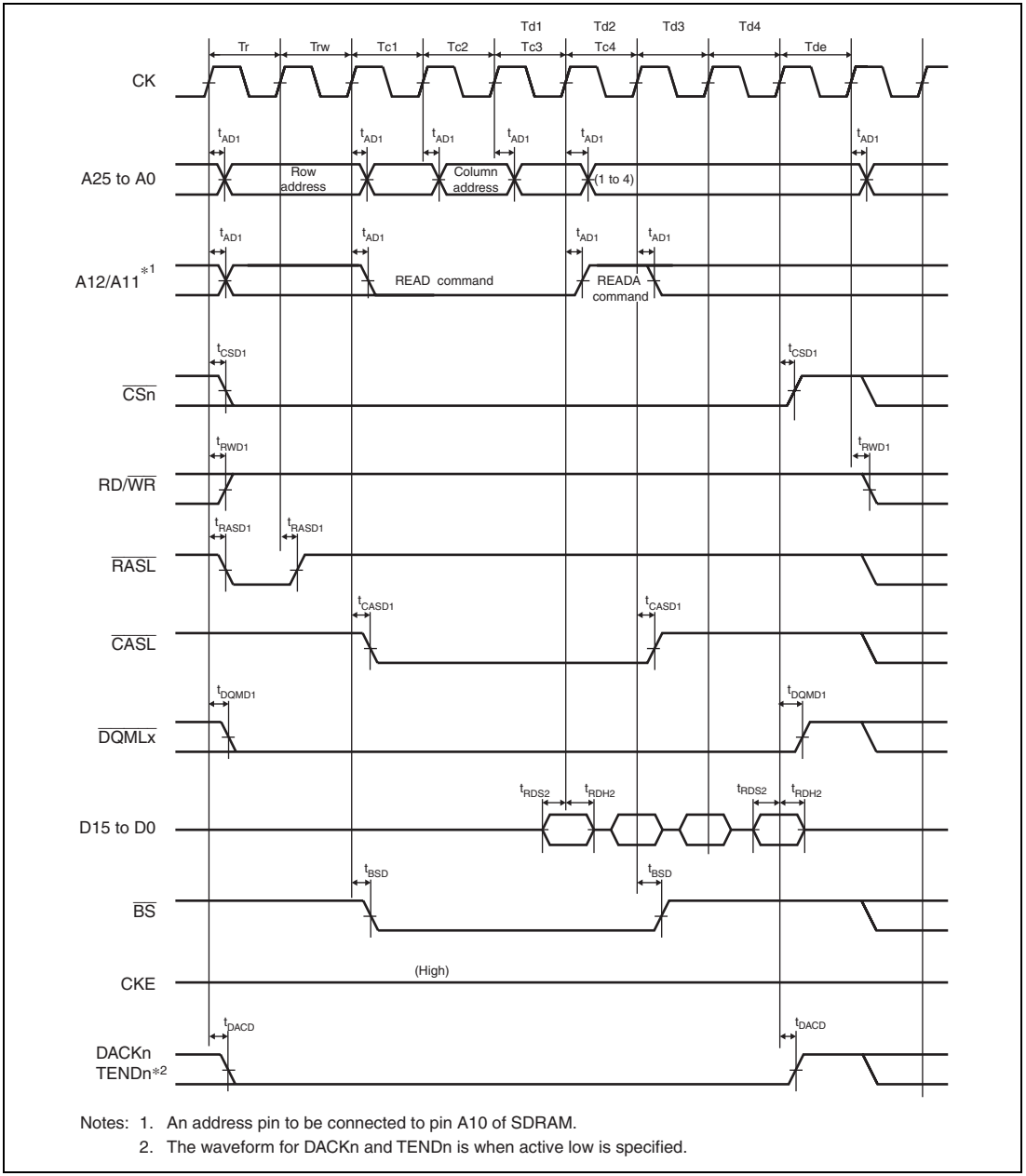
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.21 Synchronous DRAM Single Read Bus Cycle (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 1 Cycle)



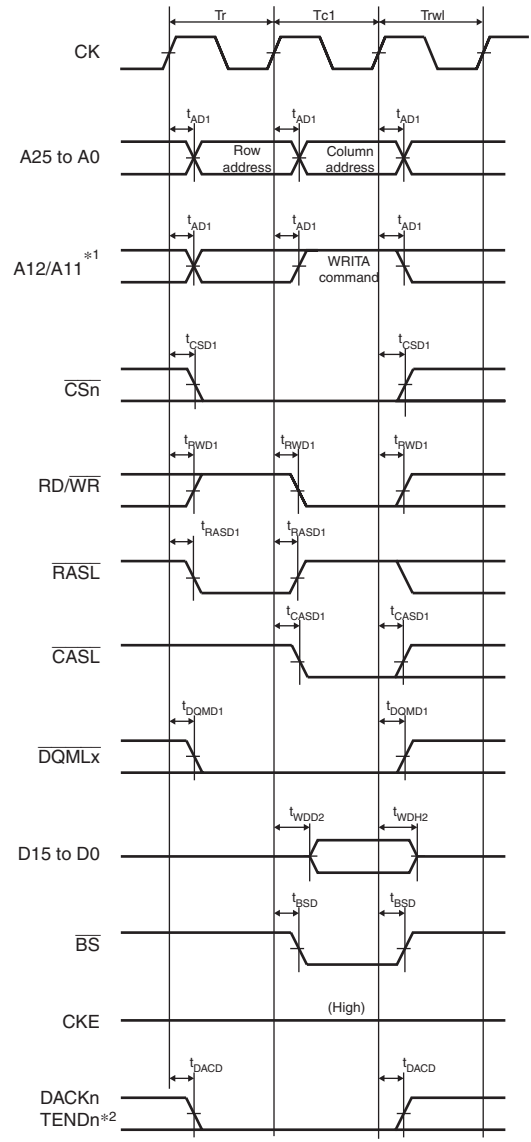
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.22 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 0 Cycle, WTRP = 1 Cycle)



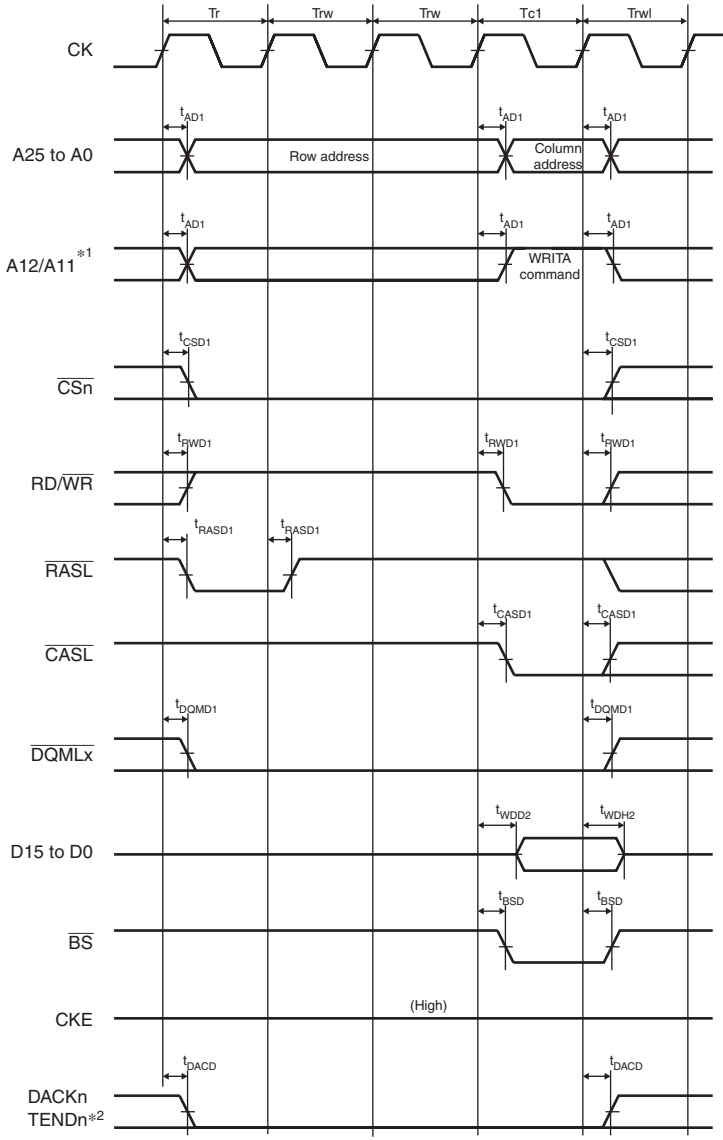
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.23 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Auto Precharge, CAS Latency 2, WTRCD = 1 Cycle, WTRP = 0 Cycle)



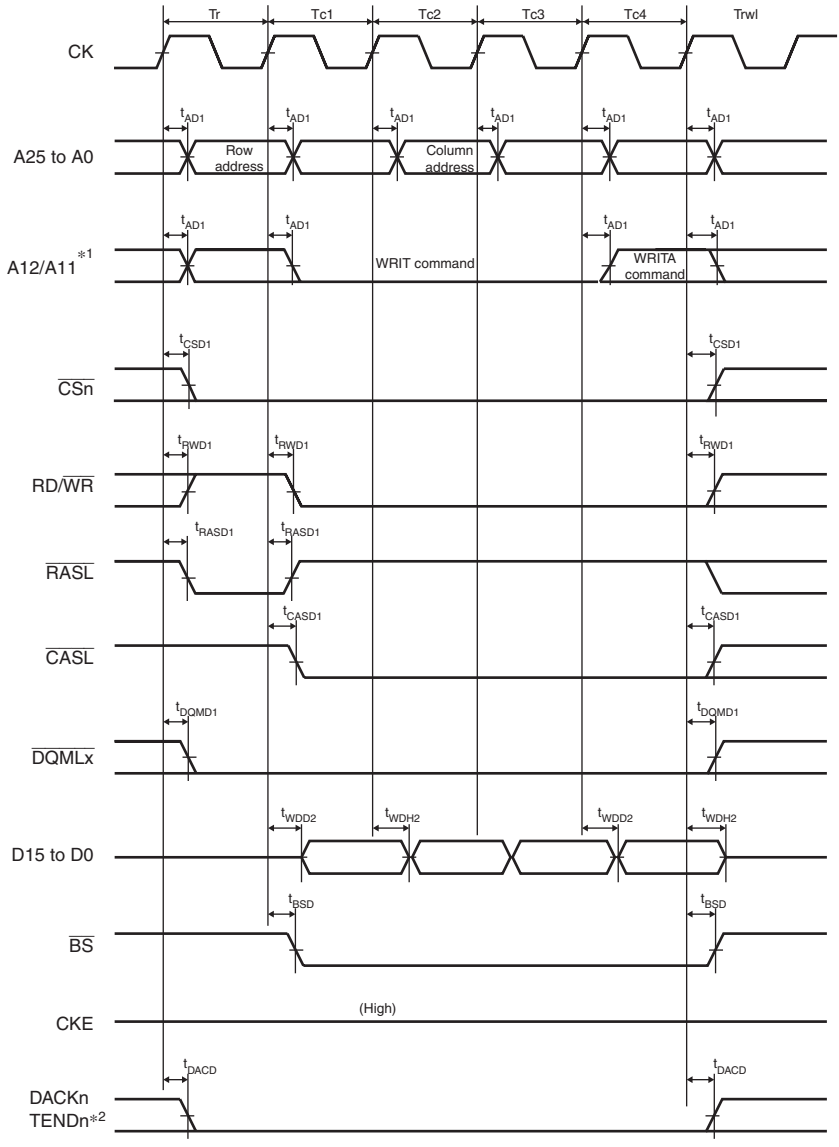
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.24 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, TRWL = 1 Cycle)



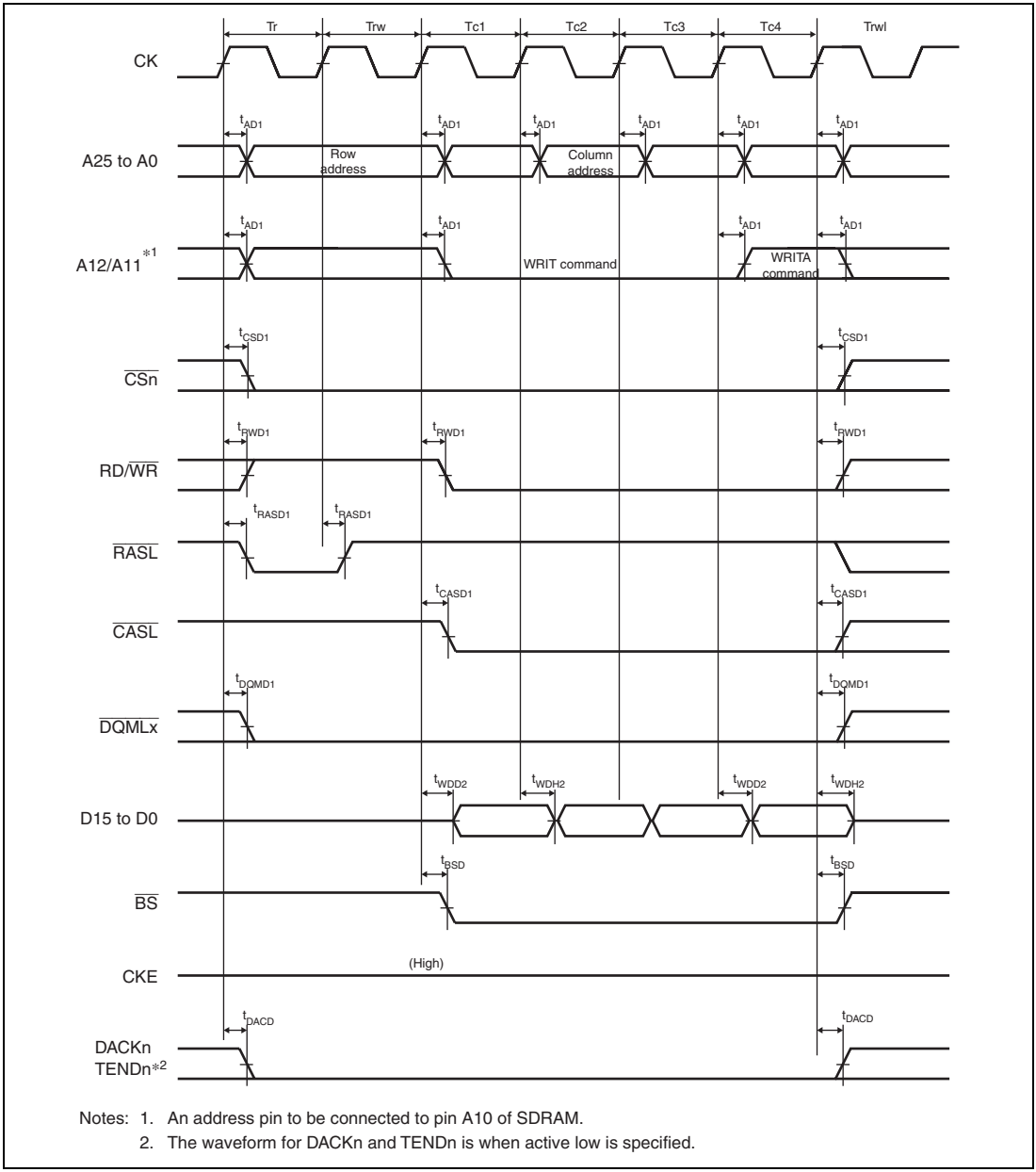
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.25 Synchronous DRAM Single Write Bus Cycle (Auto Precharge, WTRCD = 2 Cycles, TRWL = 1 Cycle)



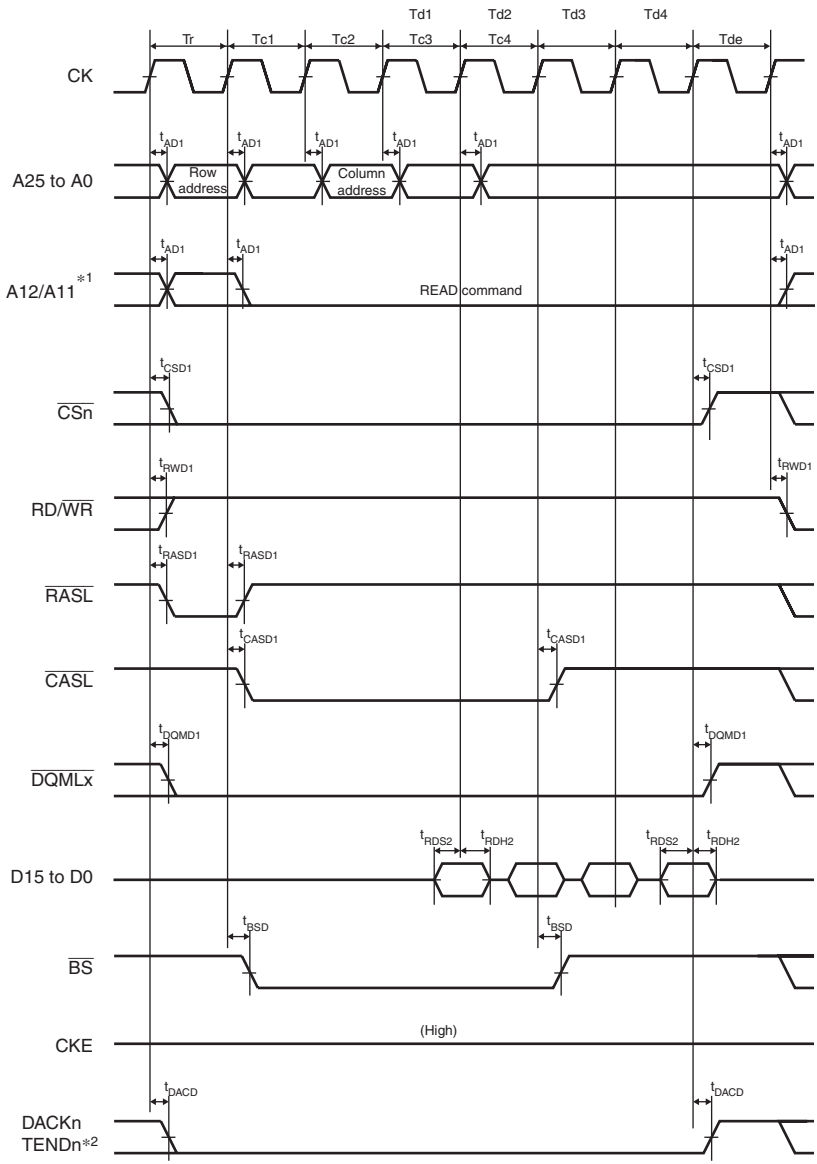
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.26 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 0 Cycle, TRWL = 1 Cycle)



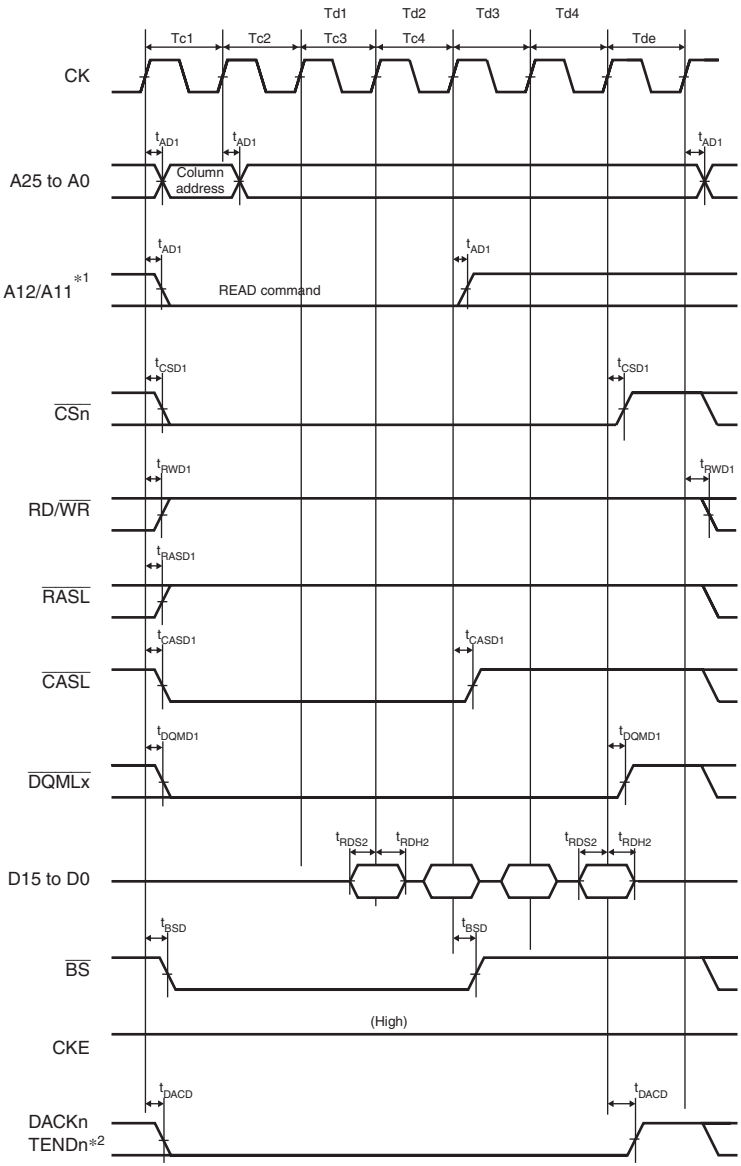
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.27 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Auto Precharge, WTRCD = 1 Cycle, TRWL = 1 Cycle)



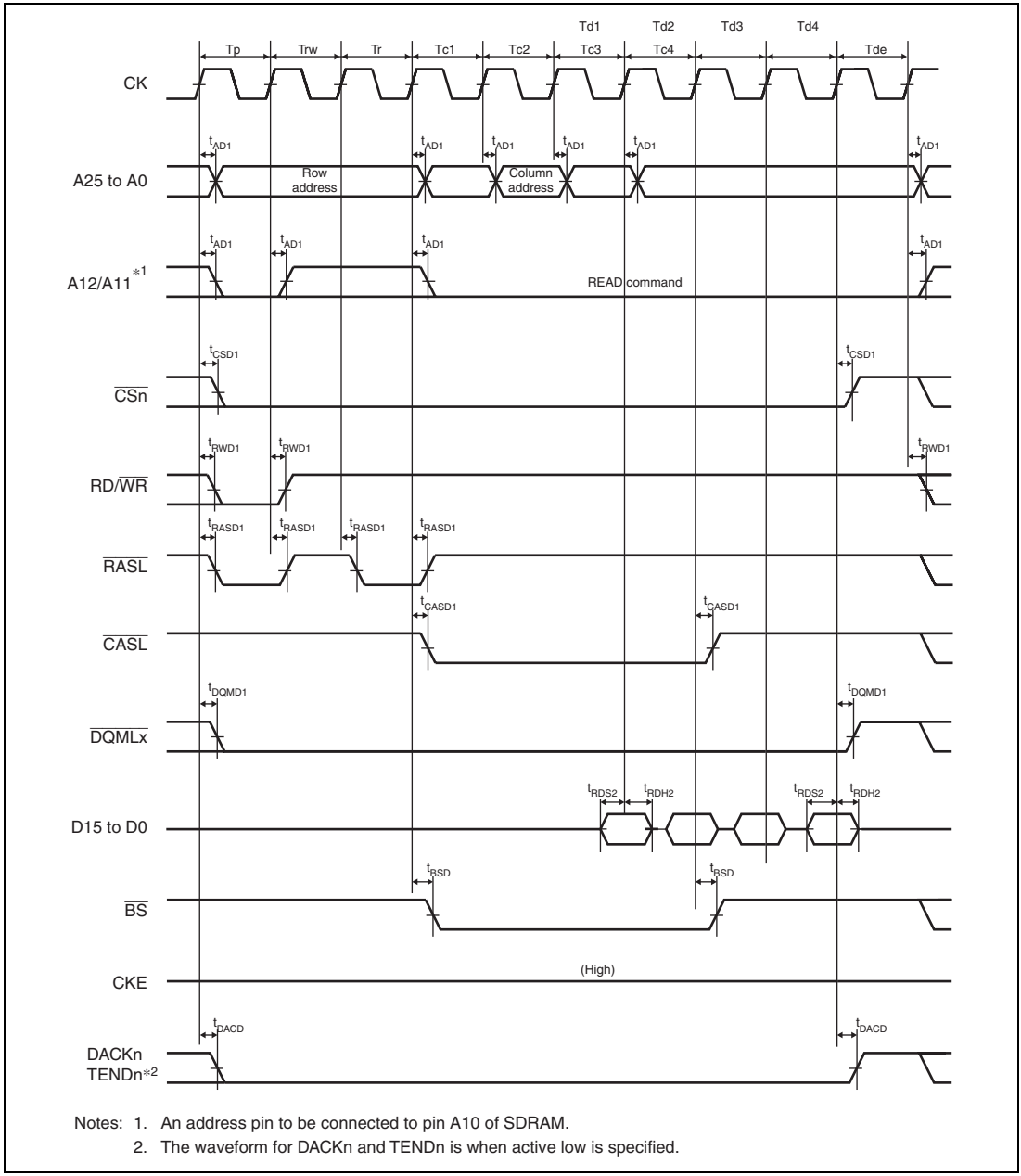
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.28 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: ACT + READ Commands, CAS Latency 2, WTRCD = 0 Cycle)



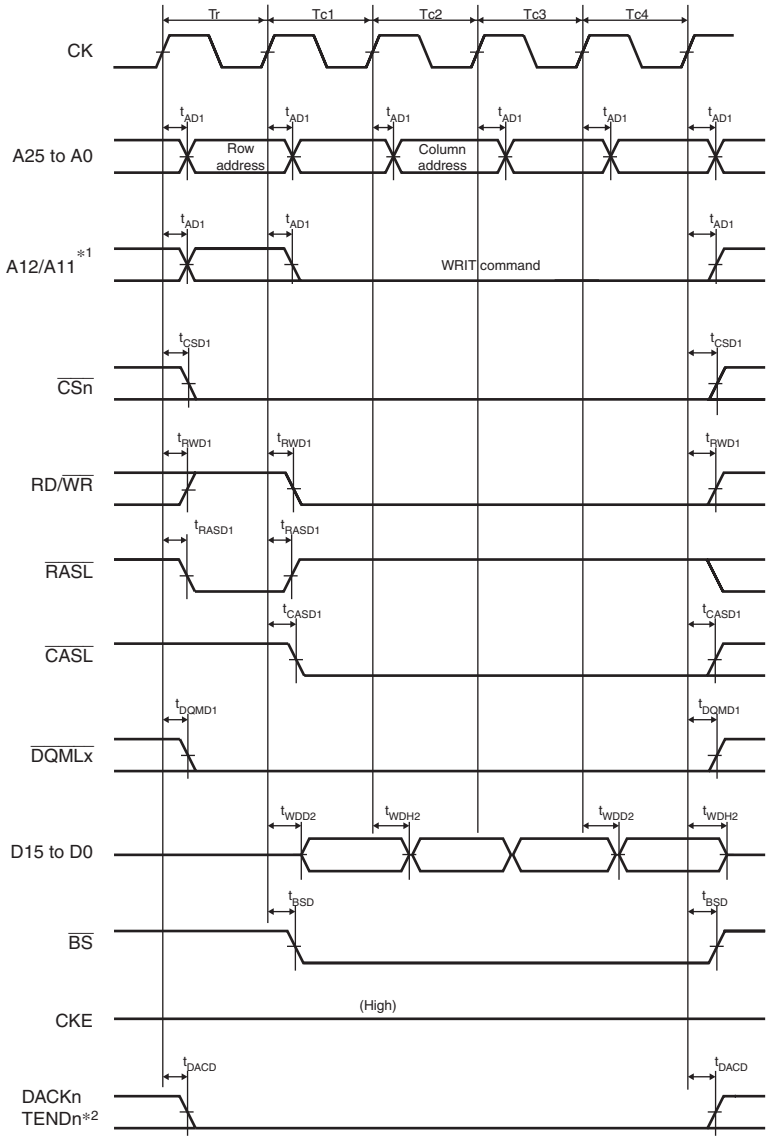
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.29 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles)
(Bank Active Mode: READ Command, Same Row Address, CAS Latency 2, WTRCD = 0 Cycle)



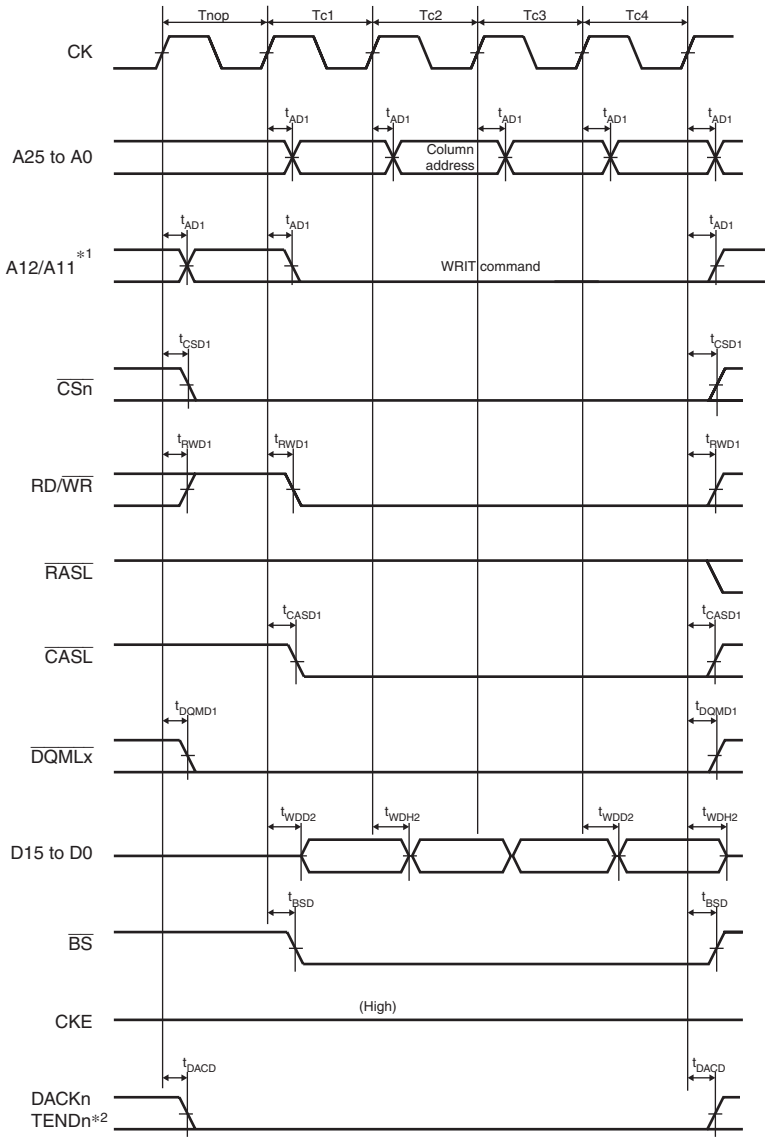
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.30 Synchronous DRAM Burst Read Bus Cycle (Four Read Cycles) (Bank Active Mode: PRE + ACT + READ Commands, Different Row Addresses, CAS Latency 2, WTRCD = 0 Cycle)



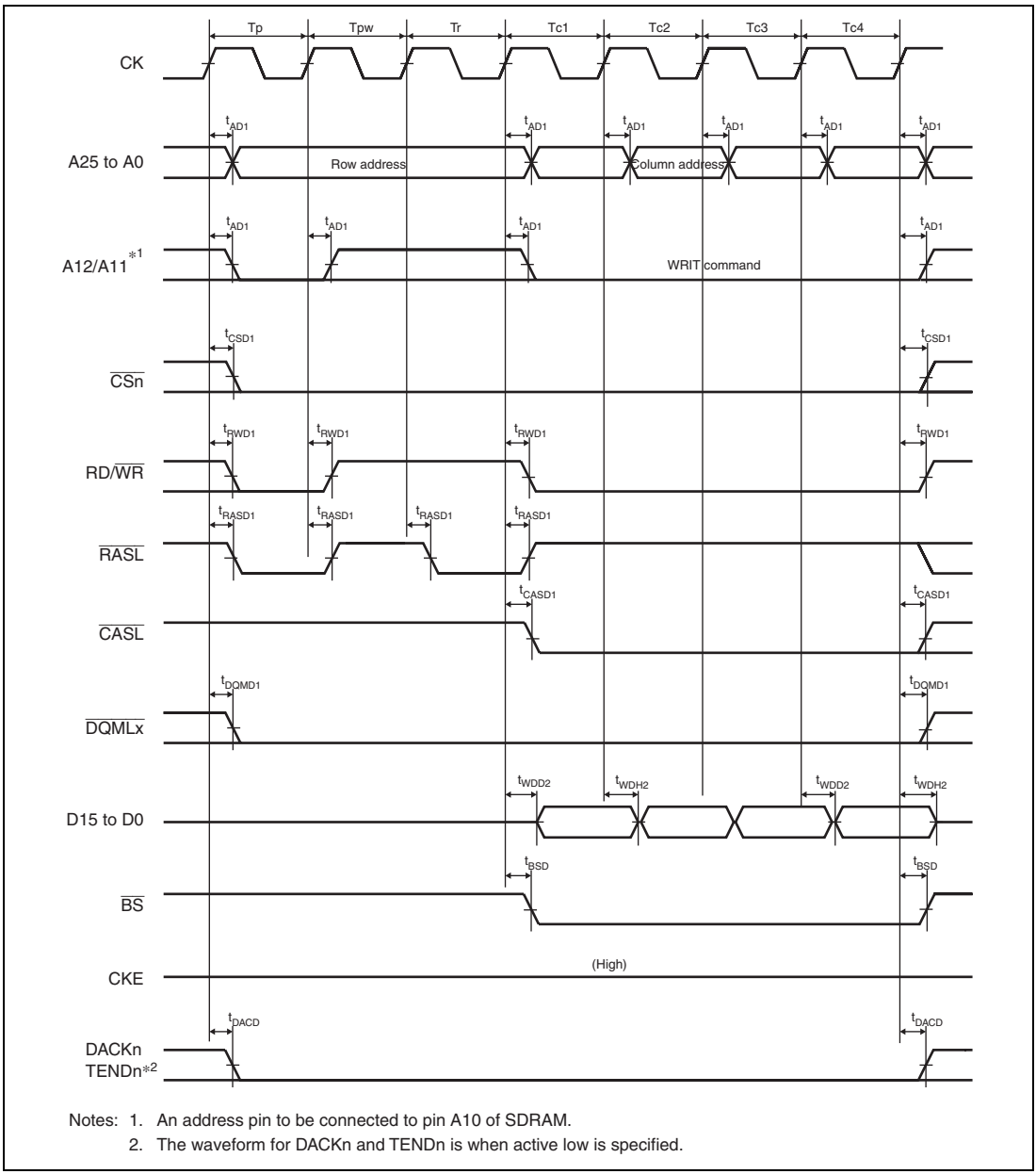
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.31 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles)
(Bank Active Mode: ACT + WRITE Commands, WTRCD = 0 Cycle, TRWL = 0 Cycle)



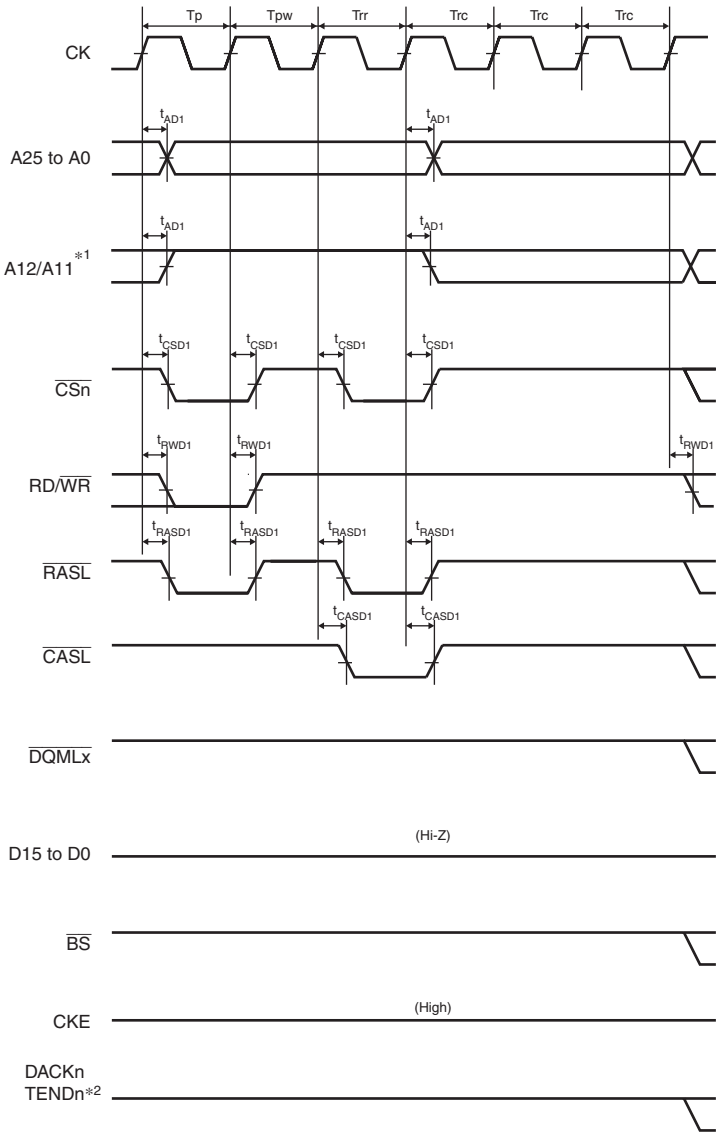
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.32 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: WRITE Command, Same Row Address, WTRCD = 0 Cycle, TRWL = 0 Cycle)



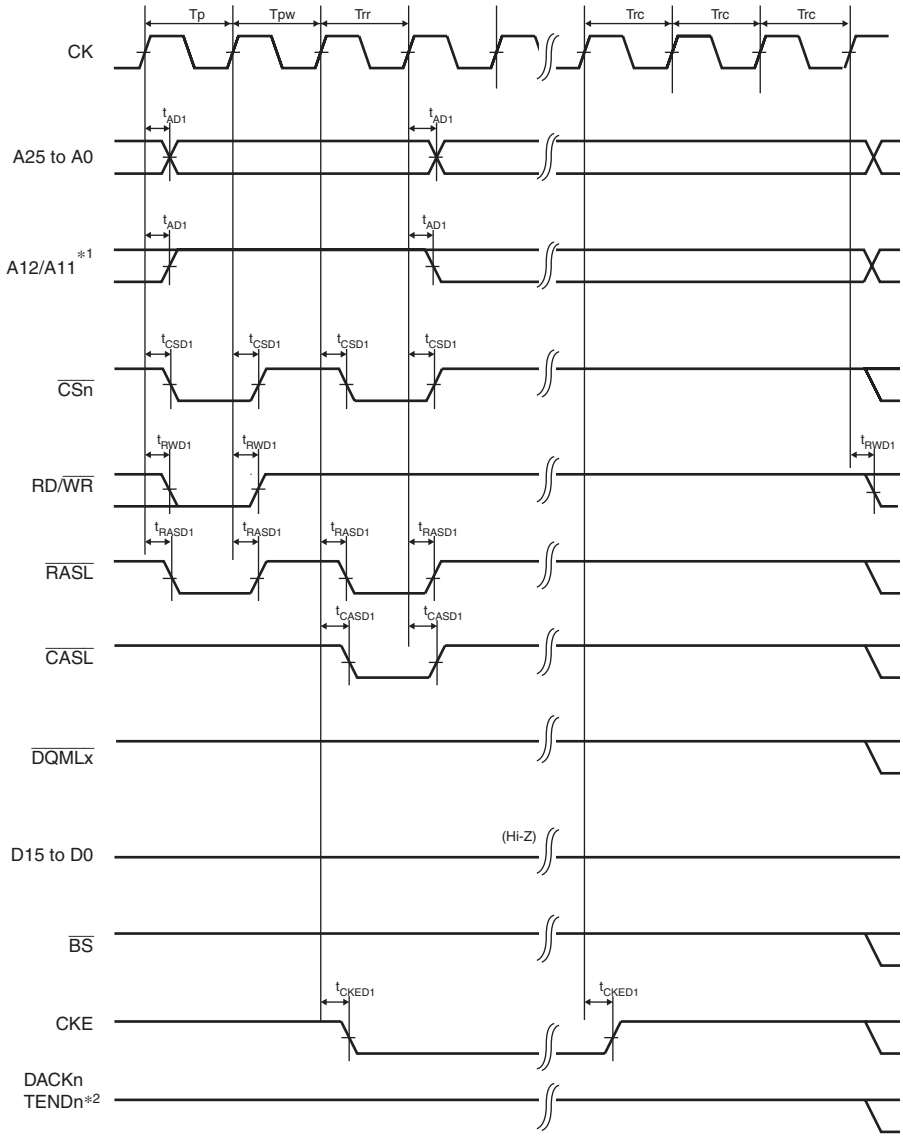
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.33 Synchronous DRAM Burst Write Bus Cycle (Four Write Cycles) (Bank Active Mode: PRE + ACT + WRITE Commands, Different Row Addresses, WTRCD = 0 Cycle, TRWL = 0 Cycle)



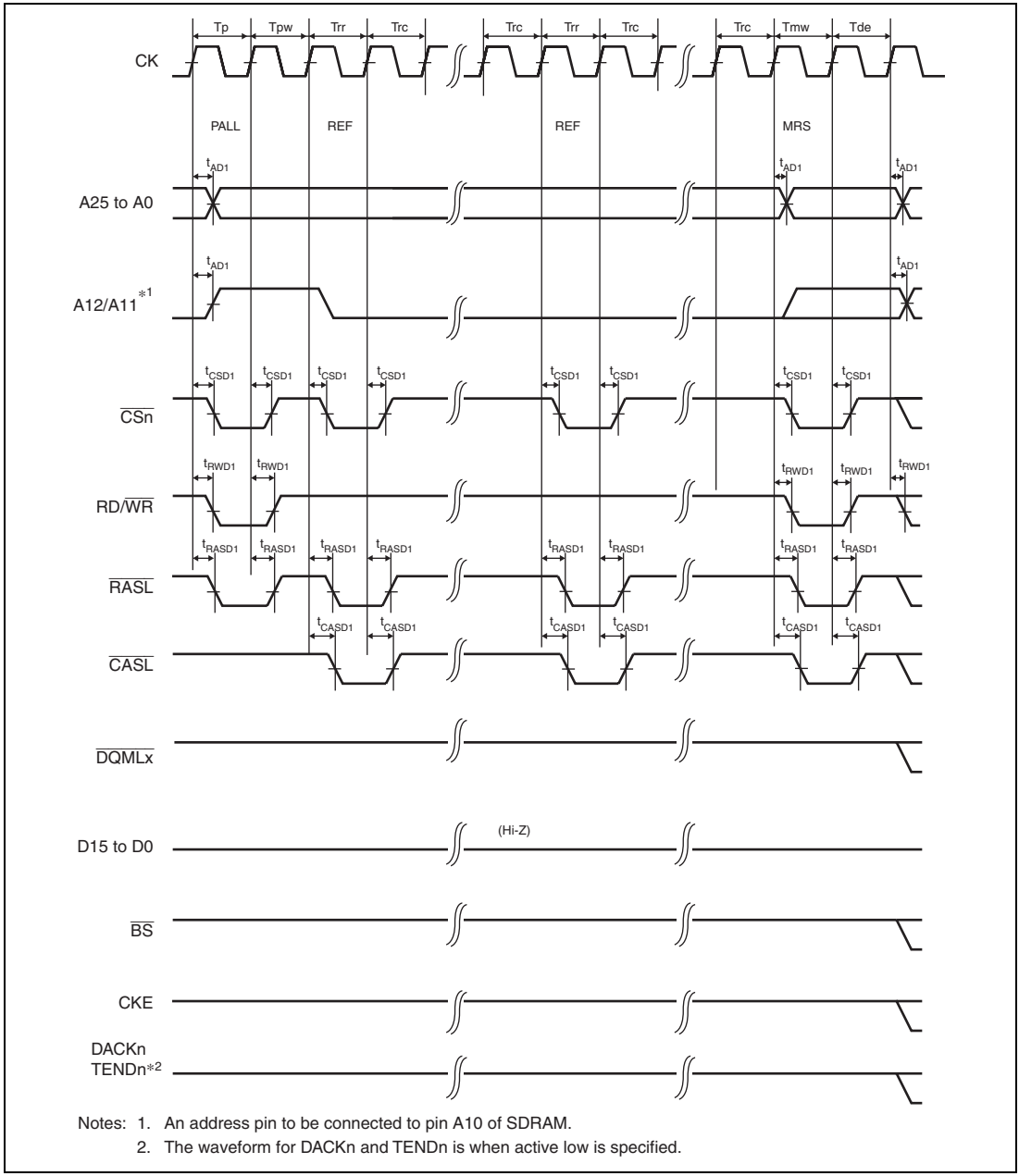
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.34 Synchronous DRAM Auto-Refreshing Timing (WTRP = 1 Cycle, WTRC = 3 Cycles)



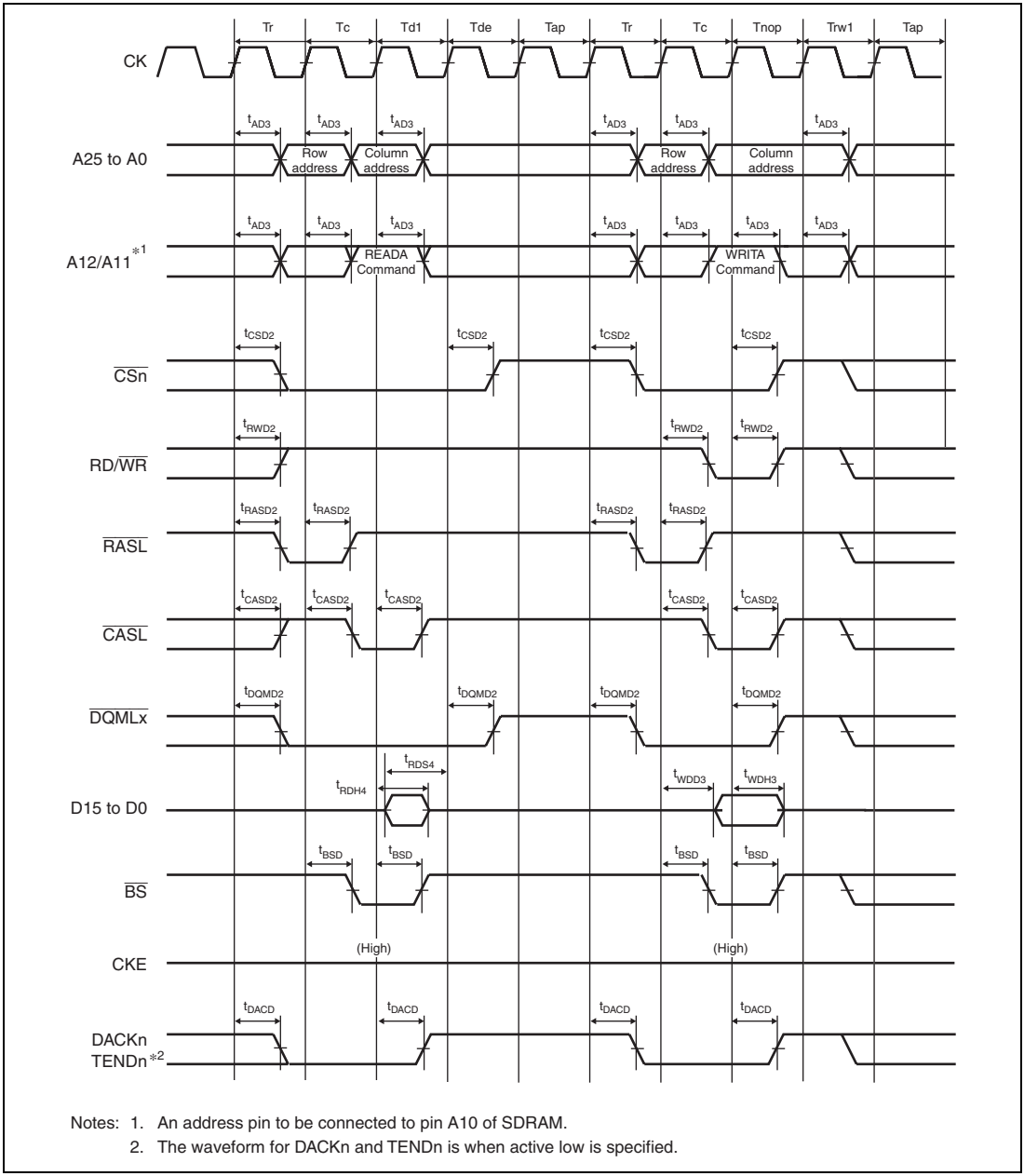
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.35 Synchronous DRAM Self-Refreshing Timing (WTRP = 1 Cycle)



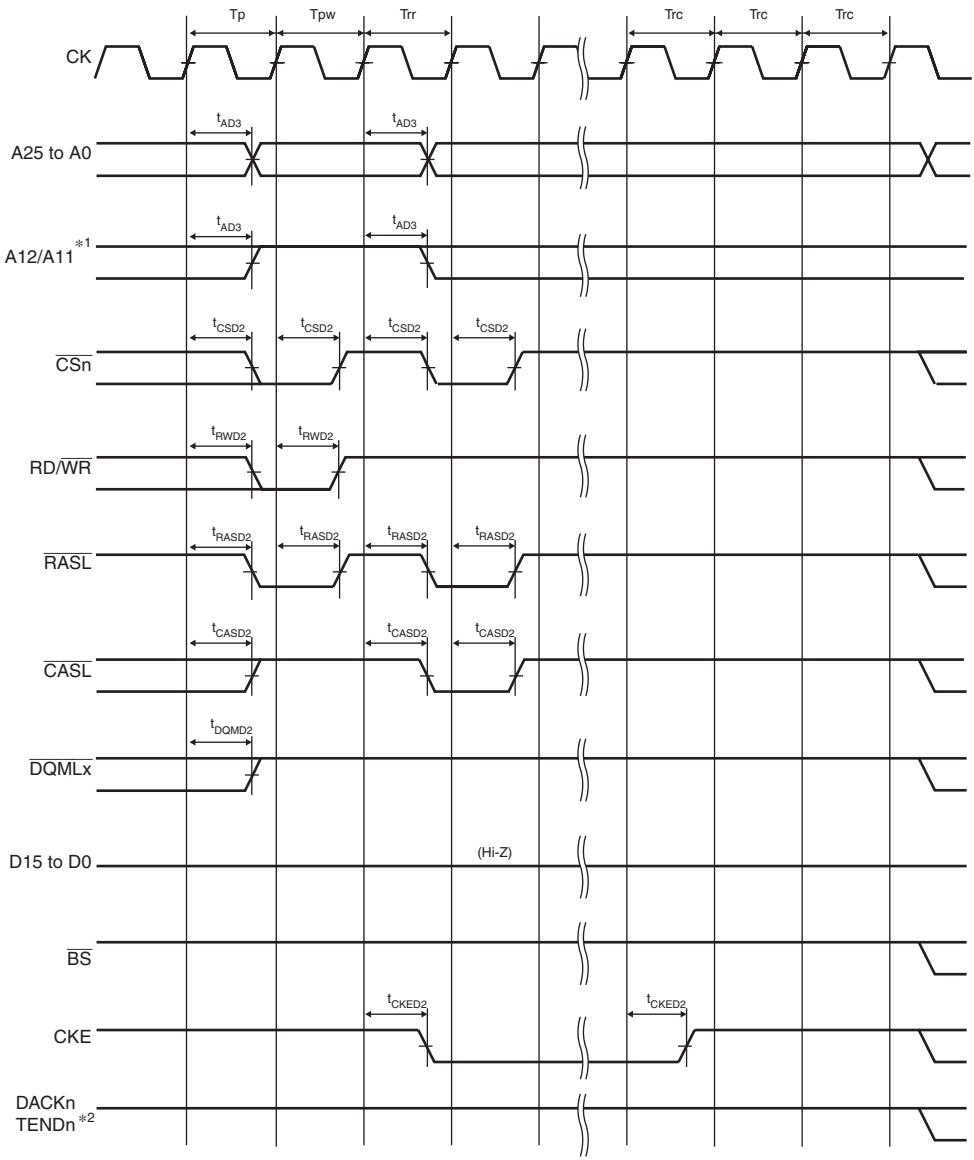
Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.36 Synchronous DRAM Mode Register Write Timing (WTRP = 1 Cycle)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.37 Synchronous DRAM Access Timing in Low-Frequency Mode (Auto-Precharge, TRWL = 2 Cycles)



Notes: 1. An address pin to be connected to pin A10 of SDRAM.
2. The waveform for DACKn and TENDn is when active low is specified.

Figure 27.38 Synchronous DRAM Self-Refreshing Timing in Low-Frequency Mode (WTRP = 2 Cycles)

27.4.4 UBC Trigger Timing

Table 27.9 UBC Trigger Timing

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6 V, Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	20	ns	Figure 27.39

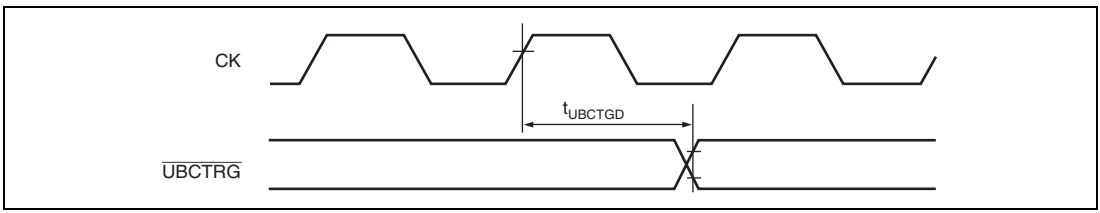


Figure 27.39 UBC Trigger Timing

27.4.5 DMAC Module Timing

Table 27.10 DMAC Module Timing

Conditions: Vcc = PLLVcc = 1.4 to 1.6 V, VccQ = 3.0 to 3.6 V, Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	20	—	ns	Figure 27.40
DREQ hold time	t_{DRQH}	20	—		
DACK, TEND delay time	t_{DACD}	—	20		Figure 27.41

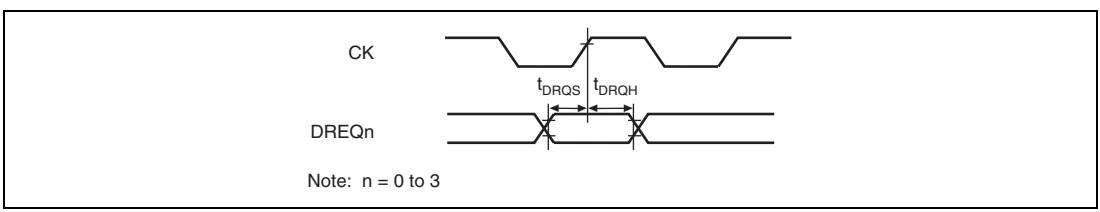
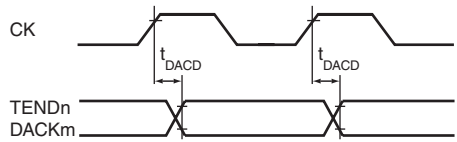


Figure 27.40 DREQ Input Timing



Note: n = 0, 1
m = 0 to 3

Figure 27.41 DACK, TEND Output Timing

27.4.6 MTU2, MTU2S Module Timing

Table 27.11 MTU2, MTU2S Module Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	50	ns	Figure 27.42
Input capture input setup time	t_{TICS}	$t_{cyc}/2 + 20$	—	ns	
Timer input setup time	t_{TCKS}	$t_{cyc} + 20$	—	ns	Figure 27.43
Timer clock pulse width (single edge)	$t_{TCKWH/L}$	1.5	—	t_{pcyc}	
Timer clock pulse width (both edges)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	
Timer clock pulse width (phase counting mode)	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

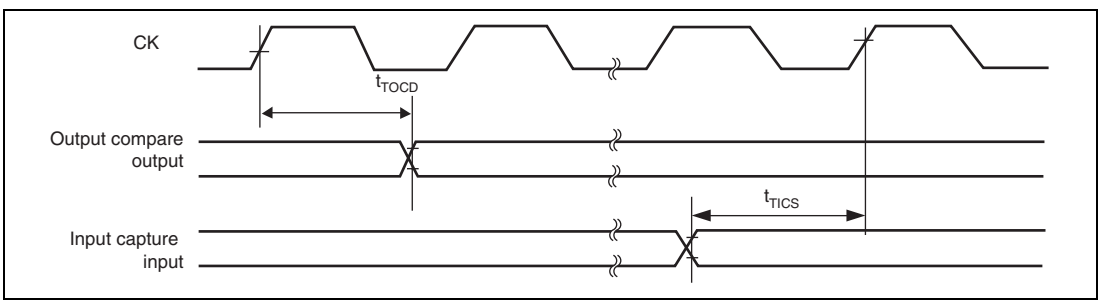


Figure 27.42 MTU2, MTU2S Input/Output Timing

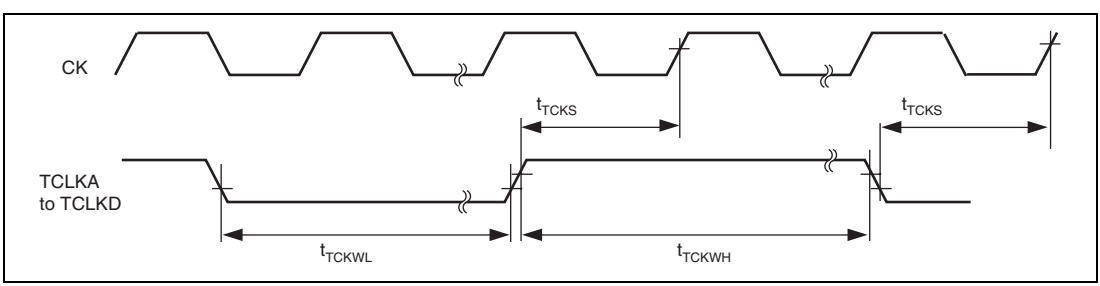


Figure 27.43 MTU2, MTU2S Clock Input Timing

27.4.7 POE2 Module Timing

Table 27.12 POE2 Module Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{POE}}$ input setup time	t_{POES}	$t_{cyc}/2 + 30$	—	ns	Figure 27.44
$\overline{\text{POE}}$ input pulse width	t_{POEW}	1.5	—	t_{pcyc}	

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

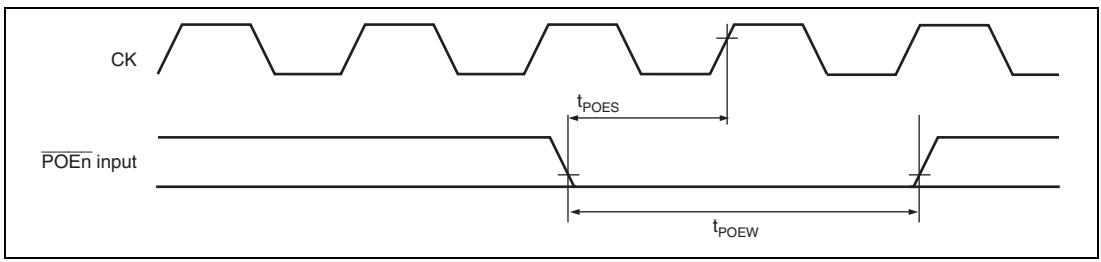


Figure 27.44 POE2 Input/Output Timing

27.4.8 Watchdog Timer Timing

Table 27.13 Watchdog Timer Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	50	ns	Figure 27.45

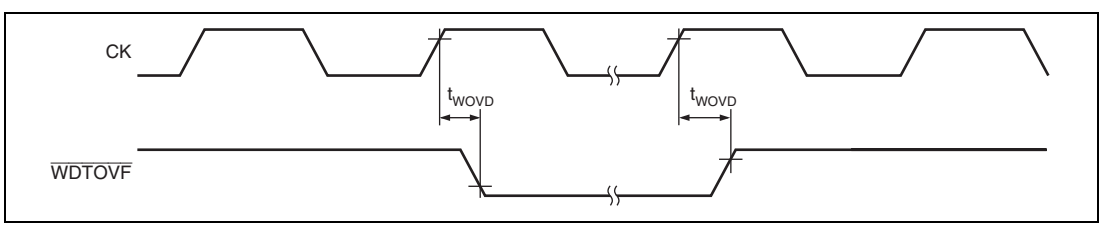


Figure 27.45 Watchdog Timer Timing

27.4.9 SCIF Module Timing

Table 27.14 SCIF Module Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(clocked synchronous)	t_{Scyc}	12	—	t_{pcyc}	Figure 27.46
	(asynchronous)		4	—	t_{pcyc}	Figure 27.46
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	Figure 27.46
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	Figure 27.46
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	Figure 27.46
Transmit data delay time (clocked synchronous)		t_{TXD}	—	100	t_{pcyc}	Figure 27.47
Receive data setup time (clocked synchronous)		t_{RXS}	100	—	ns	Figure 27.47
Receive data hold time (clocked synchronous)		t_{RXH}	100	—	ns	Figure 27.47

Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.

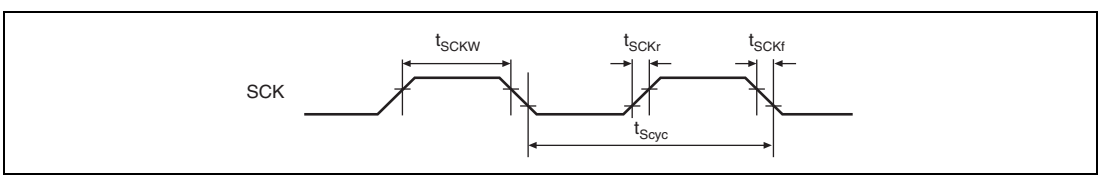


Figure 27.46 SCK Input Clock Timing

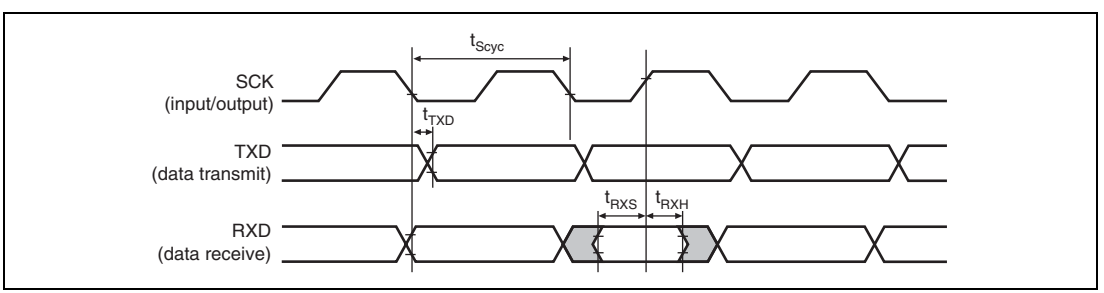


Figure 27.47 SCIF Input/Output Timing in Clocked Synchronous Mode

27.4.10 IIC3 Module Timing

Table 27.15 I²C Bus Interface 3 Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

			Specifications				
Item	**Symbol**	**Test Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**	**Figure**
SCL input cycle time	t_{SCL}		12 t_{pcyc} + 600	—	—	ns	Figure 27.48
SCL input high pulse width	t_{SCLH}		3 t_{pcyc} + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		5 t_{pcyc} + 300	—	—	ns	
SCL, SDA input rise time	t_{Sr}		—	—	300	t_{pcyc} *1	
SCL, SDA input fall time	t_{Sf}		—	—	1 t_{pcyc}	ns	
SCL, SDA input spike pulse removal time*2	t_{SP}		—	—	1 t_{pcyc}	ns	
SDA input bus free time	t_{BUF}		5	—	—	t_{pcyc} *1	
Start condition input hold time	t_{STAH}		3	—	—	t_{pcyc} *1	
Retransmit start condition input setup time	t_{STAS}		3	—	—	t_{pcyc} *1	
Stop condition input setup time	t_{STOS}		3	—	—	t_{pcyc} *1	
Data input setup time	t_{SDAS}		1 t_{pcyc} + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA capacitive load	Cb		0	—	400	pF	
SCL, SDA output fall time*3	t_{Sf}	PV_{CC} = 3.0 to 3.6 V	—	—	300	ns	

Notes: 1. t_{pcyc} indicates peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristic.

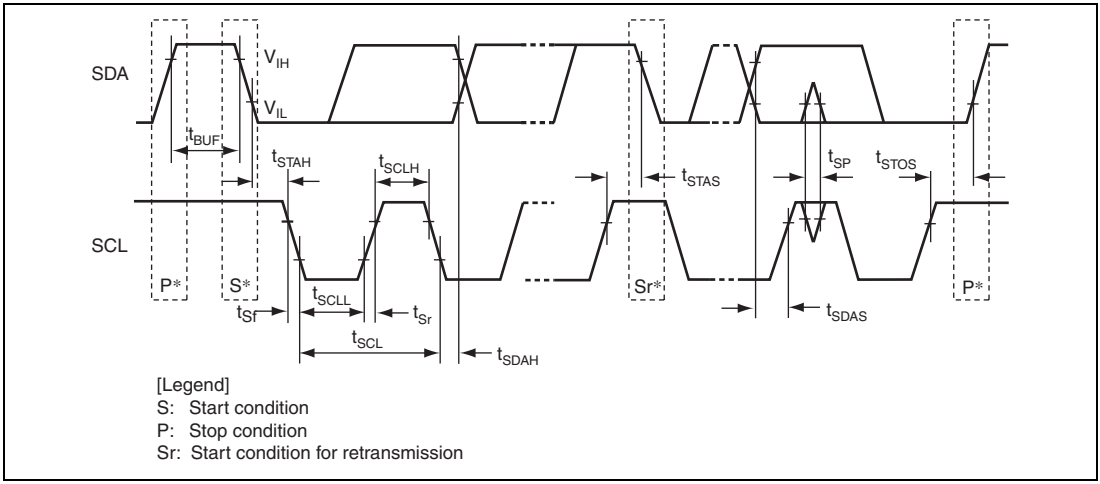


Figure 27.48 I²C Bus Interface 3 Input/Output Timing

27.4.11 A/D Trigger Input Timing

Table 27.16 A/D Trigger Input Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Module	Item		Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	B:P clock ratio = 1:1	t_{TRGS}	20	—	ns	Figure 27.49
		B:P clock ratio = 2:1		t_{cyc} + 20	—		
		B:P clock ratio = 4:1		3 × t_{cyc} + 20	—		

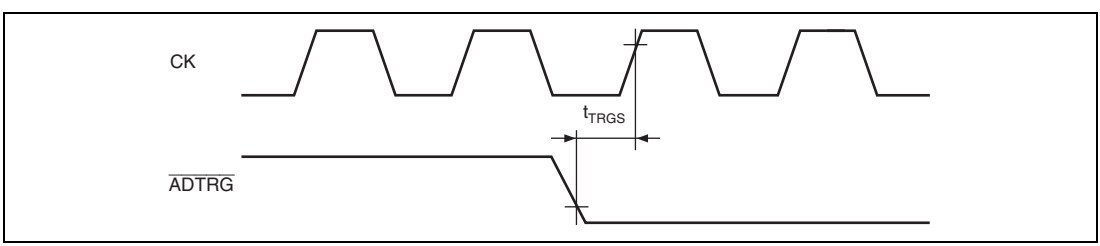


Figure 27.49 A/D Converter External Trigger Input Timing

27.4.12 I/O Port Timing

Table 27.17 I/O Port Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	100	ns	Figure 27.50
Input data setup time	t_{PORTS}	100	—		
Input data hold time	t_{PORTH}	100	—		

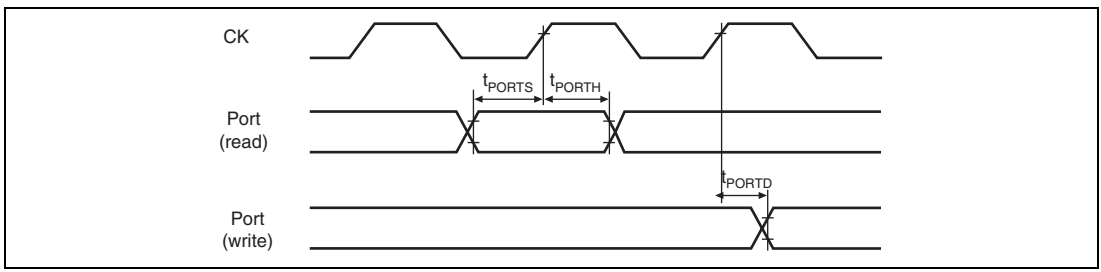


Figure 27.50 I/O Port Timing

27.4.13 H-UDI Related Pin Timing

Table 27.18 H-UDI Related Pin Timing

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VssQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Max.	Unit	Figure
TCK cycle time	t_{TCKcyc}	100*	—	ns	Figure 27.51
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 27.52
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}	—	40	ns	

Note: * Should be greater than the peripheral clock (Pϕ) cycle time.

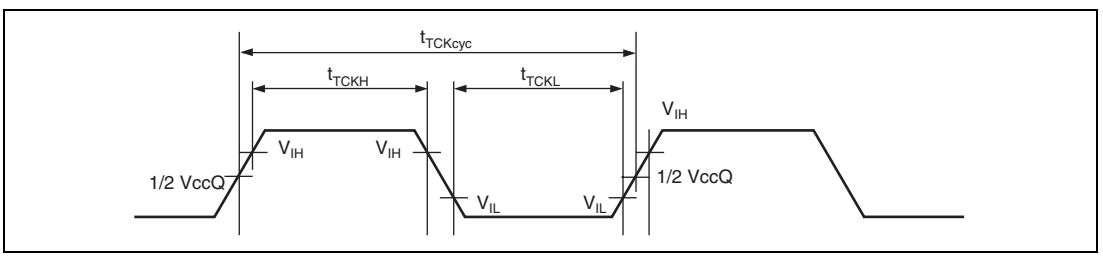


Figure 27.51 TCK Input Timing

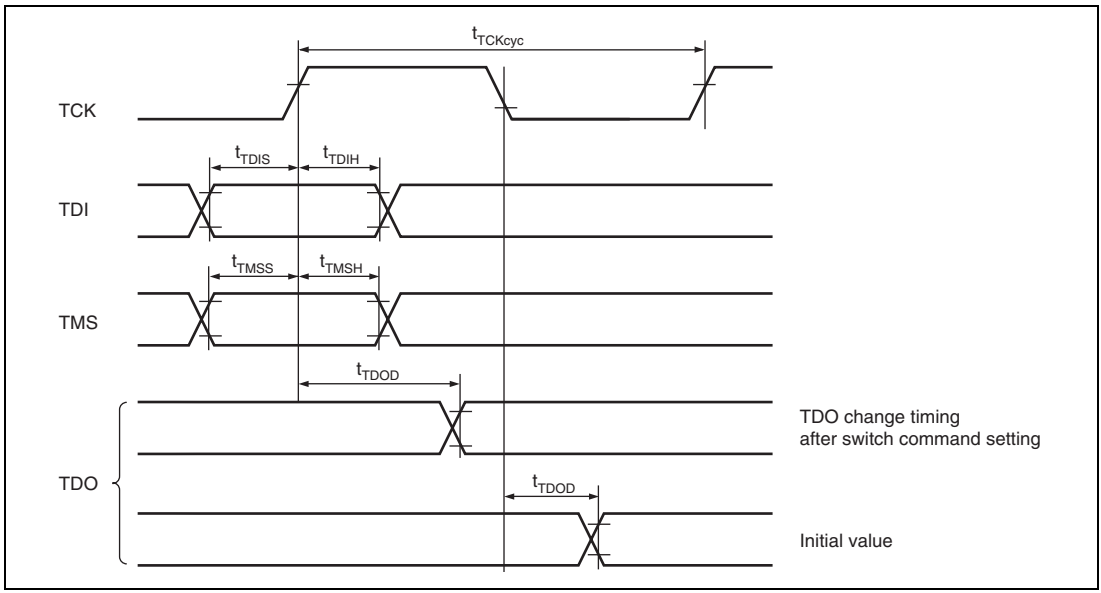
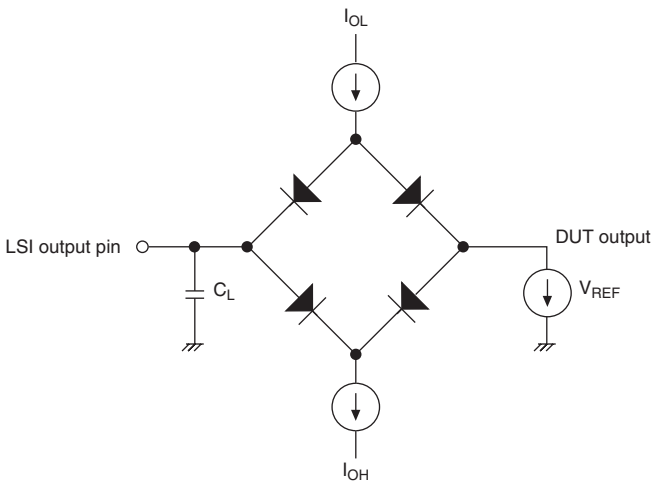


Figure 27.52 H-UDI Data Transfer Timing

27.4.14 AC Characteristics Measurement Conditions

- I/O signal reference level: VccQ/2 (VccQ = 3.0 to 3.6 V, Vcc = PLLVcc = 1.4 to 1.6 V)
- Input pulse level: VssQ to 3.0 V (where $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, MD1, MD0, MD_CLK2, MD_CLK0, $\overline{\text{ASEMD}}$, $\overline{\text{TRST}}$, and Schmitt trigger input pins are within VssQ to VccQ)
- 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:
75pF: CK
30pF: All pins
2. I_{OL} and I_{OH} are shown in table 27.4.

Figure 27.53 Output Load Circuit

27.5 A/D Converter Characteristics

Table 27.19 lists the A/D converter characteristics.

Table 27.19 A/D Converter Characteristics

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
AVcc = AV_{ref} = 4.5 V to 5.5 V,
Vss = PLLVss = VssQ = AVss = AVREFVss = 0 V, Ta = –40°C to +85°C,
V_{AN0-2} = 0.25 to AVcc –0.25 V, V_{AN3-7} = 0 V to AVcc

Item	Min.	Typ.	Max.	Unit
Resolution	—	12	—	bits
Conversion time per channel *2	1.25	—	—	µs
Analog input capacitance	—	—	5	pF
Permissible signal-source impedance	—	—	3	kΩ
Nonlinearity error	—	—	(±4.0)*1	LSB
Offset error	—	—	(±7.5)*1	LSB
Full-scale error	—	—	(±7.5)*1	LSB
Quantization error	—	—	(±0.5)*1	LSB
Absolute accuracy *3	—	—	±8.0	LSB

Notes: 1. The values in parentheses are reference values.
2. Conversion time per channel during continuous conversion. For the time from continuous conversion start to end, refer to section 17, A/D Converter (ADC).
3. The conversion error between 0 to 0.25 V of the AN0 to AN2 inputs and AVcc -0.25 V to AVcc does not meet the above value.

27.6 D/A Converter Characteristics

Table 27.20 lists the D/A converter characteristics.

Table 27.20 D/A Converter Characteristics

Conditions: Vcc = PLLVcc = 1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V, AVcc = 4.5 V to 5.5 V,
Vss = PLLVss = VssQ = AVss = AVREFVss = 0 V, Ta = –40°C to +85°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	—	bits	
Conversion time	10	—	—	μs	Load capacitance 20 pF
Absolute accuracy	—	±2.0	±3.0	LSB	Load resistance 2 MΩ
	—	—	±2.5	LSB	Load resistance 4 MΩ

27.7 Flash Memory Characteristics

Table 27.21 Flash Memory Characteristics

Conditions: Vcc = PLLVcc =1.4 V to 1.6 V, VccQ = 3.0 V to 3.6 V,
Vss = PLLVss = VccQ = 0 V, Ta = –40°C to +85°C

Item	Symbol	Min.	Typ.	Max.	Unit
Write time *1*2*4	t_P	—	2	20	ms/256 bytes
Erase time *1*2*4	t_E	—	80	260	ms/8-Kbyte block
		—	600	1600	ms/64-Kbyte block
		—	1200	3000	ms/128-Kbyte block
Write time (total) *1*2*4	Σt_P	—	4.5	12	s/512 Kbytes
Erase time (total) *1*2*4	Σt_E	—	4.5	12	s/512 Kbytes
Write and erase time (total) *1*2*4	Σt_{PE}	—	9	24	s/512 Kbytes
Number of rewrite times	N_{WEC}	100*3	—	—	times

Notes: 1. Write time and erase time depend on data.
2. Data transfer time is not included in the write and erase time.
3. Minimum value that guarantees all characteristics after rewriting (guarantees in the range from 1 to Min. value).
4. Characteristics when the number of rewrite times falls within the range including the Min. value.

Appendix

A. Pin States

Table A.1 Pin States

Pin Function		Pin State							
		Reset State					Power-Down State		Bus Master-ship Release
		Power-On*7				Manual	Software Standby	Sleep	
Type	Pin Name	Extended without ROM 8 Bits	Extended without ROM 16 Bits	Extended with ROM	Single chip				
Clock	CK (clock mode 6)	O			Z	O	O/Z*4	O	O/Z*4
	XTAL (clock mode 6)	O				O	L	O	O
	EXTAL (clock mode 6)	I				I	I	I	I
System control	$\overline{\text{RES}}$	I				I	I	I	I
	$\overline{\text{MRES}}$	Z				I	I	I	I
	$\overline{\text{WDTOVF}}$	H				O	H	O	O
	$\overline{\text{BREQ}}$	Z				I	Z	I	I
	$\overline{\text{BACK}}$	Z				O	Z	O	L
Operating mode control	MD1, MD0	I				I	I	I	I
	MD_CLK2, MD_CLK0	I				I	I	I	I
Interrupt	NMI	I				I	I	I	I
	IRQ7 to IRQ0	Z				I	I	I	I
	$\overline{\text{IRQOUT}}$	Z				O	H/Z*1	O	O
Address bus	A25 to A0	O			Z	O	O/Z*3	O	Z
Data bus	D15 to D0	Z				I/O	Z	I/O	Z

Pin Function		Pin State							
		Reset State					Power-Down State		
		Power-On*7				Manual	Software Standby	Sleep	Bus Master-ship Release
		Extended without ROM		Extended with ROM	Single chip				
Type	Pin Name	8 Bits	16 Bits						
Bus control	$\overline{\text{WAIT}}$	Z				I	Z	I	Z
	$\overline{\text{CS0}}$, $\overline{\text{CS1}}$	H		Z		O	H/Z*3	O	Z
	$\overline{\text{CS7}}$ to $\overline{\text{CS2}}$	Z				O	H/Z*3	O	Z
	$\overline{\text{BS}}$	Z				O	H/Z*3	O	Z
	$\overline{\text{RD}}$	H		Z		O	H/Z*3	O	Z
	RD/$\overline{\text{WR}}$	Z				O	H/Z*3	O	Z
	$\overline{\text{WE0}}$/$\overline{\text{DQMLL}}$	H		Z		O	H/Z*3	O	Z
	$\overline{\text{AH}}$, $\overline{\text{WE1}}$/$\overline{\text{DQMLU}}$	Z				O	H/Z*3	O	Z
	$\overline{\text{RASL}}$, $\overline{\text{CASL}}$	Z				O	O/Z*2	O	O/Z*2
	CKE	Z				O	O/Z*2	O	O/Z*2
	$\overline{\text{REFOUT}}$	Z				O	H/Z*1	O	O
DMAC	DREQ3 to DREQ0	Z				I	Z	I	I
	DACK3 to DACK0	Z				O	O/Z*1	O	O
	TEND1, TEND0	Z				O	O/Z*1	O	O
MTU2	TCLKA, TCLKB, TCLKC, TCLKD	Z				I	Z	I	I
	TIOC0A*6, TIOC0B*6, TIOC0C*6, TIOC0D*6	Z				I/O	K/Z*1	I/O	I/O
	TIOC1A, TIOC1B	Z				I/O	K/Z*1	I/O	I/O
	TIOC2A, TIOC2B	Z				I/O	K/Z*1	I/O	I/O
	TIOC3A, TIOC3B*6, TIOC3C, TIOC3D*6	Z				I/O	K/Z*1	I/O	I/O

Pin Function		Pin State							
		Reset State					Power-Down State		
		Power-On*7							
		Extended without ROM		Extended with ROM	Single chip	Manual	Software Standby	Sleep	Bus Mastership Release
Type	Pin Name	8 Bits	16 Bits						
MTU2	TIOC4A*6, TIOC4B*6, TIOC4C*6, TIOC4D*6	Z				I/O	K/Z*1	I/O	I/O
	TIC5U, TIC5V, TIC5W	Z				I	Z	I	I
MTU2S	TIOC3AS, TIOC3BS*6, TIOC3CS, TIOC3DS*6	Z				I/O	K/Z*1	I/O	I/O
	TIOC4AS*6, TIOC4BS*6, TIOC4CS*6, TIOC4DS*6	Z				I/O	K/Z*1	I/O	I/O
	TIC5US, TIC5VS, TIC5WS	Z				I	Z	I	I
POE2	$\overline{\text{POE8}}$, $\overline{\text{POE7}}$, $\overline{\text{POE4}}$, $\overline{\text{POE3}}$, $\overline{\text{POE1}}$, $\overline{\text{POE0}}$	Z				I	Z	I	I
SCIF	SCK3 to SCK0	Z				I/O	K/Z*1	I/O	I/O
	RXD3 to RXD0	Z				I	Z	I	I
	TXD3 to TXD0	Z				O	O/Z*1	O	O
WAVE	WSCK	Z				O	O/Z*1	O	O
	WRXD	Z				I	Z	I	I
	WTXD	Z				O	O/Z*1	O	O
A/D converter	AN7 to AN0	Z				I	Z	I	I
	$\overline{\text{ADTRG}}$	Z				I	Z	I	I
D/A converter	DA1, DA0	Z				O	O	O	O
IIC3	SCL	Z				I/O	Z	I/O	I/O
	SDA	Z				I/O	Z	I/O	I/O

Pin Function		Pin State							
		Reset State					Power-Down State		
		Power-On*7				Manual	Software Standby	Sleep	Bus Master-ship Release
		Extended without ROM		Extended with ROM	Single chip				
Type	Pin Name	8 Bits	16 Bits						
Emulator	AUDSYNC	—				O	O	O	O
	AUDCK	—				O	O	O	O
	AUDATA3 to AUDATA0	—				O	O	O	O
	$\overline{\text{ASEMD}}$	I				I	I	I	I
	$\overline{\text{ASEBRK}}$/ $\overline{\text{ASEBRKAK}}$	O				O	I	O	O
	$\overline{\text{TRST}}$	I				I	I	I	I
	TCK	I				I	I	I	I
	TDI	I				I	I	I	I
	TDO	O/Z*5				O/Z*5	O/Z*5	O/Z*5	O/Z*5
	TMS	I				I	I	I	I
UBC	$\overline{\text{UBCTRG}}$	Z				O	O/Z*1	O	O
I/O port	PA25 to PA0	Z				I/O	K/Z*1	I/O	I/O
	PB30 to PB22, PB21 to PB18*6, PB19 to PB12 PB13 to PB10*6, PB9, PB8, PB7 to PB4*6, PB3 to PB0	Z				I/O	K/Z*1	I/O	I/O
	PD15 to PD0	Z				I/O	K/Z*1	I/O	I/O
	PF1, PF0	Z				I	Z	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. Controlled by the HIZ bit in standby control register 3 (STBCR3) (see section 23, Power-Down Modes).
2. Controlled by the HIZCNT bit in the common control register of the BSC (see section 8, Bus State Controller (BSC)).
3. Controlled by the HIZMEM bit in the common control register of the BSC (see section 8, Bus State Controller (BSC)).
4. Controlled by the HIZCKIO bit in the common control register of the BSC (see section 8, Bus State Controller (BSC)).
5. Z when the TAP controller of the H-UDI is neither the Shift-DR nor Shift-IR state.
6. High-impedance control through POE2 (see section 12, Port Output Enable 2 (POE2)).
7. Power-on reset by low-level input to the $\overline{\text{RES}}$ pin. The pin states after a power-on reset by the H-UDI reset assert command or WDT overflow are the same as the initial pin states at normal operation (see section 19, Pin Function Controller (PFC)).

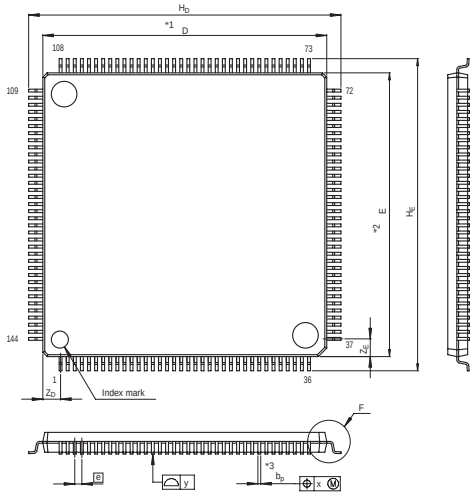
B. Product Lineup

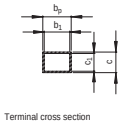
Table B.1 Product Lineup

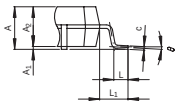
Product Type	Product Code	Package
SH7211	R5F72115D160FPV	LQFP2020-144Cu (FP-144LV)

C. Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP144-20x20-0.50	PLQP0144KA-A	144P6Q-A / FP-144L / FP-144LV	1.2g







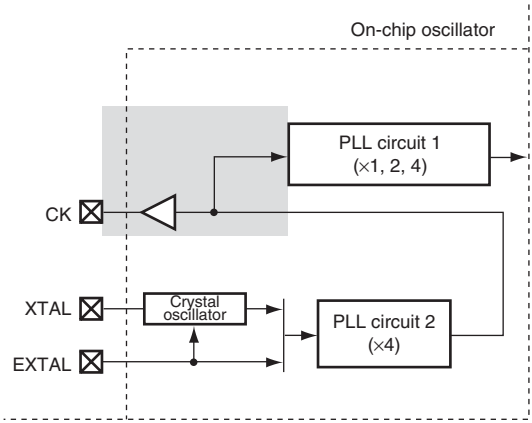
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure C.1 FP-144LV

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
1.1 SH7211 Features Table 1.1 SH7211 Features	5	Table amended **Items** / **Specification** Multi-function timer pulse unit 2 (MTU2): • Pulse output modes Toggle, PWM, complementary PWM, and reset-synchronized PWM modes
	6	Table amended **Items** / **Specification** User break controller (UBC): • Four break channels • Addresses, 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 • Realtime branch trace Advanced user debugger II (AUD- II): • 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
1.4 Pin Functions Table 1.2 Pin Functions	9	Table amended **Classification** / **Symbol** / **I/O** / **Name** / **Function** Clock / CK / O / System clock / Supplies the system clock to external devices.
3.1 Selection of Operating Modes Table 3.1 Selection of Operating Modes	55	Table amended **Pin Setting** (**FWE**, **MD1**, **MD0**) / **Mode No.** / **Mode Name** / **On-Chip ROM** / **Bus Width of CS0 Space SH7211F** Mode 6* / 1 / 1 / 0 / User program mode / Active / Set by CS0BCR in BSC
4.1 Features	61	Description amended • Clock operating modes Either the internal crystal resonator or the input on the external clock-signal line can be selected.

Item	Page	Revision (See Manual for Details)
4.1 Features	62	Figure amended
Figure 4.1 Block Diagram of Clock Pulse Generator		
(4) Divider 1	63	Description amended Divider 1 generates a clock signal at the operating frequency used by the internal clock (Iϕ), the bus clock (Bϕ), the peripheral clock (Pϕ), the MTU2S clock (Mϕ), or the AD clock (Aϕ).
4.2 Input/Output Pins	65	Table amended
Table 4.1 Pin Configuration and Functions of the Clock Pulse Generator		**Pin Name** / **Symbol** / **I/O** / **Function (Clock Operating Mode 6)** Clock output pin / CK / Output / Clock output pin. This pin can be high impedance.

Item	Page	Revision (See Manual for Details)
4.4.1 Frequency Control Register (FRQCR)	70	Table amended

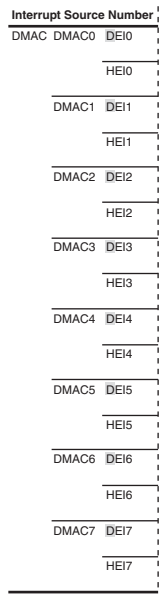
Bit	Bit Name	Initial Value	R/W	Description
6 to 4	IFC[2:0]	000	R/W	Internal Clock (Iϕ) Frequency Division Ratio These bits specify the frequency division ratio of the internal clock with respect to the output frequency of PLL circuit 1. If a prohibited value is specified, correct operation cannot be guaranteed. 000: × 1 time 001: × 1/2 time 011: × 1/4 time Other than above: Setting prohibited
2 to 0	PFC[2:0]	011	R/W	Peripheral Clock (Pϕ) Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock with respect to the output frequency of PLL circuit 1. If a prohibited value is specified, correct operation cannot be guaranteed. 000: × 1 time 001: × 1/2 time 011: × 1/4 time 101: × 1/8 time Other than above: Setting prohibited

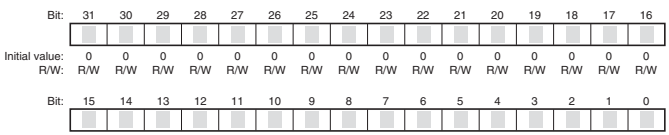
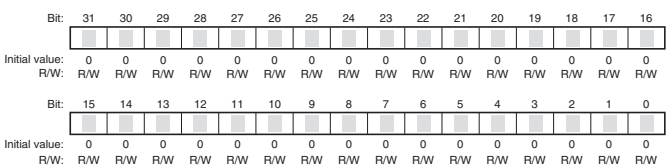
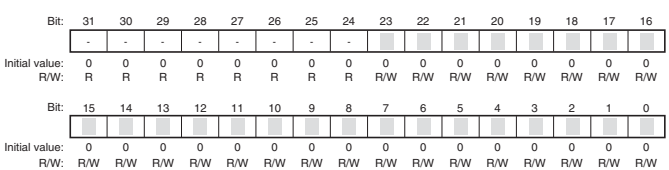
5.2.2 Power-On Reset (1) Power-On Reset by Means of $\overline{\text{RES}}$ Pin	84	Description amended When the $\overline{\text{RES}}$ pin is driven low, this LSI enters the power-on reset state. To reliably reset this LSI, the $\overline{\text{RES}}$ pin should be kept at the low level for the duration of the oscillation settling time at power-on or when in software standby mode (when the clock is halted), or at least 20-tcyc when the clock is running.
5.2.3 Manual Reset (1) Manual Reset by Means of $\overline{\text{MRES}}$ Pin	86	Description amended 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-tcyc .

Item	Page	Revision (See Manual for Details)
5.3.1 Address Error Sources Table 5.6 Bus Cycles and Address Errors	87	Table amended

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

5.3.2 Address Error Exception Handling	88	Note added 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: Note: * This is the case in which an address error was caused by data read or write. When an address error is caused by an instruction fetch, and if the bus cycle in which the address error occurred does not end by step 3 above, the CPU restarts the address error exception handling until the bus cycle ends.
6.3.4 IRQ Interrupt Request Register (IRQRR)	110	Note added Note: * Only 0 can be written to clear the flag after 1 is read.

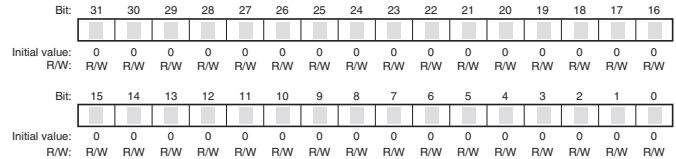
Item	Page	Revision (See Manual for Details)
6.5 Interrupt Exception Handling Vector Table and Priority Table 6.4 Interrupt Exception Handling Vectors and Priorities	119	Table amended 
6.7 Interrupt Response Time Table 6.5 Interrupt Response Time	130	Notes amended Notes: 2. In the case that (Iϕ, Bϕ, Pϕ) = (160 MHz, 40 MHz, 40 MHz).
8.1 Features Figure 8.1 Block Diagram of BSC	177	Description amended [Legend] CMNCR: Common control register CSnWCR: CSn space wait control register (n = 0 to 7) CSnBCR: CSn space bus control register (n = 0 to 7)
8.3.2 Setting Operating Modes	181	Description amended • Initial Settings of Endianness … In on-chip ROM-enabled mode, all the endianness of areas 0 to 7 can be changed by register settings in the program. Little endian cannot be selected in area 0. Since both 32-bit and 16-bit accesses are included in instruction fetches, no instructions can be assigned in little endian area. Accordingly, instructions should be executed in big endian area.

Item	Page	Revision (See Manual for Details)
8.4.2 CSn Space Bus Control Register (CSnBCR) (n = 0 to 7)	186	Note amended Note: * CSnBCR samples the external pins (MD1 and MD0) that specify the bus width at power-on reset.
8.5.6 SDRAM Interface Table 8.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)-1	249	Note 3 deleted
Table 8.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and Address Multiplex Output (3)-2	250	Note 3 deleted
8.5.12 Others (3) On-Chip Peripheral Module Access	295-297	Description replaced
9.3.1 DMA Source Address Registers (SAR)	307	Figure amended 
9.3.2 DMA Destination Address Registers (DAR)	308	Figure amended 
9.3.3 DMA Transfer Count Registers (DMATCR)	309	Figure amended 

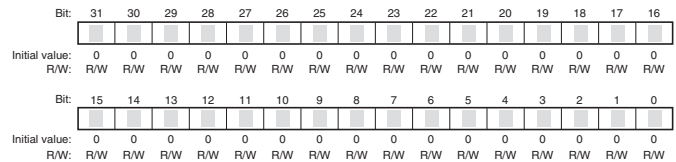
Item	Page	Revision (See Manual for Details)
9.3.4 DMA Channel Control Registers (CHCR)	313	Table amended

Bit	Bit Name	Initial Value	R/W	Descriptions
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

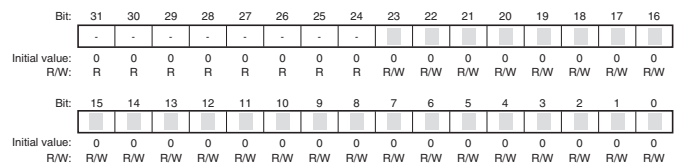
9.3.5 DMA Reload Source Address Registers (RSAR)	318	Figure amended



9.3.6 DMA Reload Destination Address Registers (RDAR)	319	Figure amended



9.3.7 DMA Reload Transfer Count Registers (RDMATCR)	320	Figure amended



10.1 Features	353	Table amended
Table 10.1 MTU2 Functions		

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DMAC 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	—

Item	Page	Revision (See Manual for Details)
10.3.3 Timer I/O Control Register (TIOR) Table 10.28 TIORU_5, TIORV_5, and TIORW_5 (Channel 5)	387	Table amended

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
				1	Input capture register	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

10.3.20 Timer Output Control Register 1 (TOCR1)	421	Notes amended Notes: 1. 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. 2. Setting the TOCL bit to 1 prevents accidental modification when the CPU goes out of control. 3. Clearing the TOCS0 bit to 0 makes this bit setting valid.
10.3.23 Timer Gate Control Register (TGCR)	428	Note added

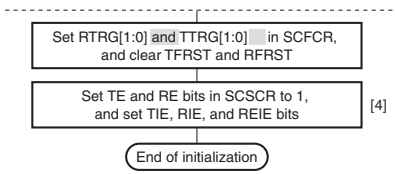
Bit	Bit Name	Initial Value	R/W	Description
3	FB*	0	R/W	External Feedback Signal Enable This bit selects whether the switching of the output of the positive/reverse phase is carried out automatically with the MTU2/channel 0 TGRA, TGRB, TGRC input capture signals or by writing 0 or 1 to bits 2 to 0 in 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).

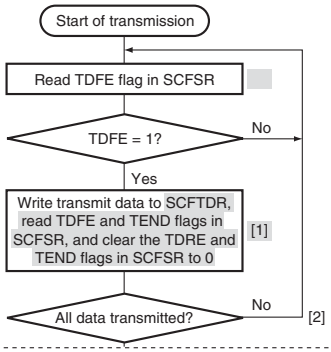
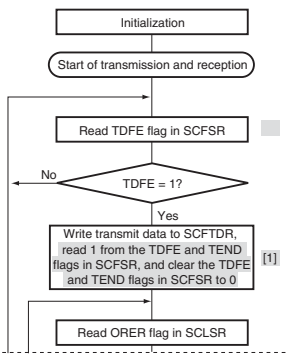
Note: * If the BDC bit in the MTU2S is set to 1, the FB bit should not be cleared to 0.

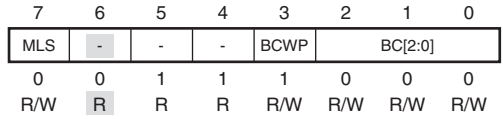
10.4.8 Complementary PWM Mode Figure 10.41 Example of Operation without Dead Time	482	Figure replaced

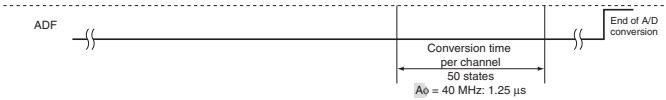
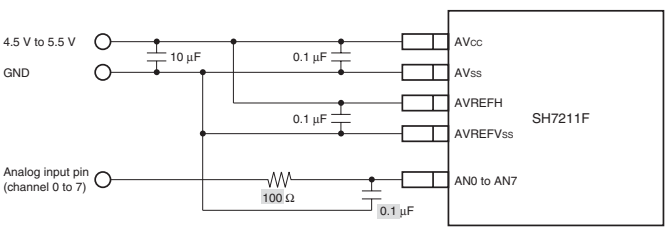
Item	Page	Revision (See Manual for Details)
10.4.8 Complementary PWM Mode (3) Interrupt Skipping in Complementary PWM Mode (c) Buffer Transfer Control Linked with Interrupt Skipping	513	Description amended … While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. There are two types of timing in which data is transferred from the buffer register to the temporary register or to general register, depending on the buffer register modification timing after an interrupt occurrence. Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR).
Figure 10.77 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)	515	Figure replaced
Figure 10.78 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period	516	Figure replaced
10.4.11 External Pulse Width Measurement Figure 10.88 Example of External Pulse Width Measurement (Measuring High Pulse Width)	528	Figure replaced
10.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection	565	Description added … 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 additionally supports the function that can capture TCNT_1 and TCNT_2 simultaneously via a single input capture input. This function allows 32-bit counter fetches without TCNT_1 and TCNT_2 capture timing deviation. For details, see section 10.3.8, Timer Input Capture Control Register (TICCR).
12.6.1 Pin Status When the WDT Issues a Power-On Reset	635	Description replaced

<table>
<tr><th>Item</th><th>Page</th><th>Revision (See Manual for Details)</th></tr>
<tr><td>15.1 Features</td><td>666</td><td>Description deleted
• 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.</td></tr>
<tr><td>15.3.7 Serial Status Register (SCFSR)</td><td>682</td><td>Table amended
<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>5</td><td>TDFE</td><td>1</td><td>R/(W)*</td><td>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.
1: The quantity of transmit data in SCFTDR is equal to or 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 equal to or 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.</td></tr>
</table></td></tr>
<tr><td>15.3.8 Bit Rate Register (SCBRR)</td><td>687</td><td>Description amended
The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset. Each channel has independent baud rate generator control, so different values can be set in four channels.</td></tr>
</table>

Item	Page	Revision (See Manual for Details)
15.3.8 Bit Rate Register (SCBRR) • Asynchronous mode:	687	Description amended (1) In normal mode (when the ABCS bit in SCSEMR is 0) (2) In serial extended mode (when the ABCS bit in SCSEMR is 1)
	688	Description amended (1) In normal mode (when the ABCS bit in SCSEMR is 0) (2) In serial extended mode (when the ABCS bit in SCSEMR is 1)
Table 15.4 Bit Rates and SCBRR Settings (Asynchronous Mode)	689	Table replaced
Table 15.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode)	690	Table replaced
Table 15.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)	691	Table replaced
Table 15.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)		Table replaced
Table 15.8 Maximum Bit Rates with External Clock Input (Clocked Synchronous Mode, $t_{Scyc} = 12t_{pcyc}$)		Table replaced
15.3.12 Line Status Register (SCLSR)	697	Note added Note: * Only 0 can be written to clear the flag after 1 is read.
15.4.2 Operation in Asynchronous Mode Figure 15.3 Sample Flowchart for SCIF Initialization	704	Figure amended 

Item	Page	Revision (See Manual for Details)
15.4.3 Operation in Clocked Synchronous Mode Figure 15.11 Sample Flowchart for Transmitting Serial Data	714	Figure amended  [1] SCIF status check and transmit data write: Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, read 1 from the TDFE and TEND flags, and clear these flags to 0. [2] Serial transmission continuation procedeure: To continue serial transmission, read 1 from the TDFE flag to confirm that writing is possible, them write data to SCFTDR, and then clear the TDFE flag to 0.
Figure 15.16 Sample Flowchart for Transmitting/Receiving Serial Data	719	Figure amended  [1] SCIF status check and transmit data write: Read SCFSR and check that the TDFE flag is set to 1, then write transmit data to SCFTDR, read 1 from the TDFE and TEND flags, and clear these flags to 0. The transition of the TDFE flag from 0 to 1 can also be identified by a TXI interrupt. [2] Receive error handling: Read the ORER flag in SCLSR 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. [3] SCIF status check and receive data read:
15.6.7 FER and PER Flags in the Serial Status Register (SCFSR)	724	Newly added
16.3.1 I^2C Bus Control Register 1 (ICCR1)	729	Table amended Bit 6: Bit Name RCVD; Initial Value 0; R/W R/W Description: Reception Disable Enables or disables continuous reception when TRS = 0 and ICDRR is not read. If ICDRR cannot be read by the rising of 8th clock cycle of SCL in master receive mode, reception in byte units should be performed by setting the RCVD bit to 1. 0: Enables continuous reception 1: Disables continuous reception
Table 16.3 Transfer Rate	731	Table replaced

Item	Page	Revision (See Manual for Details)
16.3.3 I^2C Bus Mode Register (ICMR)	734	Figure amended  Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 6 / — / 0 / R / Reserved This bit is always read as 0. The write value should always be 0.
16.4.2 Master Transmit Operation	745	Description amended 1. Set the ICE bit in ICCR1 to 1. Also, set the bits CKS[3:0] in ICCR1. (Initial setting)
16.6 Bit Synchronous Circuit Figure 16.22 Bit Synchronous Circuit Timing	765	Figure replaced
Table 16.5 Time for Monitoring SCL	766	Table amended **CKS3** / **CKS2** / **Time for Monitoring SCL** 0 / 0 / 9 tpcyc* 1 / 21 tpcyc* 1 / 0 / 33 tpcyc* 1 / 81 tpcyc* Note: * tpcyc stands for the peripheral clock (Pϕ) cycle.
16.7 Usage Notes	767	Newly added
17.1 Features	769	Description amended • High-speed conversion When Aϕ = 40 MHz: Minimum 1.25 μs per channel AD clock = 40 MHz, 50 conversion states • A/D data registers Eight A/D data registers (ADDR) are provided. A/D conversion results are stored in A/D data registers (ADDR) that correspond to the input channels.

Item	Page	Revision (See Manual for Details)
17.4.3 Input Sampling and A/D Conversion Time Figure 17.4 A/D Conversion Timing (Single-Cycle Scan Mode)	786	Figure amended 
17.7.4 Notes on Board Design	793	Description amended Also, digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference power supply (AVREF), the analog power supply (AV_{CC}), the analog ground (AV_{SS}), and the analog reference ground ($AVREFV_{SS}$). Also, AV_{SS} should be connected at one point to a stable digital ground (V_{SS}) on the board.
17.7.5 Notes on Noise Countermeasures Figure 17.8 Example of Analog Input Pin Protection Circuit	794	Figure amended 
17.7.7 Treatment of AV_{CC} and AV_{SS} When the A/D Converter is Not Used	794	Newly added
18.5.3 Setting Analog Input Voltage	801	Description amended 1. AV_{CC} and AV_{SS} input voltages Input voltages AV_{CC} and AV_{SS} should be $V_{CC} \leq AV_{CC} \leq 5.0$ V ± 0.5 V and $AV_{SS} = V_{SS}$. Do not leave the AV_{CC} and AV_{SS} pins open when the A/D converter or D/A converter is not in use and in software standby mode. When not in use, connect AV_{CC} to the power supply ($V_{CC}Q$) and AV_{SS} to the ground ($V_{SS}Q$). 2. Setting range of AVREF input voltage Set the voltage range of the AVREF pin as AVREF = AV_{CC} ± 0.3 V when the A/D converter or D/A converter is used, or as AVREF = AV_{CC} when no A/D converter or D/A converter is used.

Item	Page	Revision (See Manual for Details)
19.1 Register Descriptions Table 19.5 Register Configuration	810	Table amended

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
IRQOUT function control register	IFCR	R/W	H'0000	H'FFFE38A2	16

19.1.2 Port A Control Registers H1 to H3, L1 to L4 (PACRH1 to PACRH3, PACRL1 to PACRL4) (4) Port A Control Register L4 (PACRL4)	819	Table amended

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PA15MD[2:0]	000*	R/W	PA15 Mode Select the function of the PA15/A15/SCK3 pin. • Area 0: 16-bit mode/8-bit mode 000: PA15 I/O (port) 001: A15 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: SCK3 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited

(5) Port A Control Register L3 (PACRL3)	821	Table amended

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PA10MD[2:0]	000*	R/W	PA10 Mode Select the function of the PA10/A10 pin. • Area 0: 16-bit mode/8-bit mode 000: PA10 I/O (port) 001: A10 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

(7) Port A Control Register L1 (PACRL1)	826	Table amended

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PA1MD[2:0]	000*	R/W	PA1 Mode Select the function of the PA1/A1 pin. • Area 0: 16-bit mode/8-bit mode 000: PA1 I/O (port) 001: A1 output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Item	Page	Revision (See Manual for Details)
19.1.4 Port B Control Registers H1 to H4, L1 to L4 (PBCRH1 to PBCRH4, PBCRL1 to PBCRL4) (1) Port B Control Register H4 (PBCRH4)	829	Table amended Bit: 2 to 0 Bit Name: PB28MD[2:0] Initial Value: 000 R/W: R/W Description: PB28 Mode Select the function of the PB28/DACK0/TIOC1A/RXD3 pin. 000: PB28 I/O (port) 001: Setting prohibited 010: Dack0 output (DMAC) 011: Setting prohibited 100: Setting prohibited 101: TIOC1A I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
(2) Port B Control Register H3 (PBCRH3)	831	Table amended Bit: 2 to 0 Bit Name: PB24MD[2:0] Initial Value: 000 R/W: R/W Description: PB24 Mode Select the function of the PB24/TEND1/IRQ2/TCLKB/RXD3/AUDATA3 pin. 000: PB24 I/O (port) 001: Setting prohibited 010: TEND1 output (DMAC) 011: IRQ2 input (INTC) 100: TCLKB input (MTU2) 101: RXD3 input (SCIF) 110: AUDATA3 output (AUD) 111: Setting prohibited.
(3) Port B Control Register H2 (PBCRH2)	833	Table amended Bit: 2 to 0 Bit Name: PB20MD[2:0] Initial Value: 000 R/W: R/W Description: PB20 Mode Select the function of the PB20/$\overline{BS}$/TIOC3DS pin. 000: PB20 I/O (port) 001: $\overline{BS}$ output (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC3DS I/O (MTU2S) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Item	Page	Revision (See Manual for Details)
19.1.4 Port B Control Registers H1 to H4, L1 to L4 (PBCRH1 to PBCRH4, PBCRL1 to PBCRL4) (4) Port B Control Register H1 (PBCRH1)	835	Table amended Note added Note: * The initial value depends on the operating mode of the LSI.
(5) Port B Control Register L4 (PBCRL4)	837	Table amended
(6) Port B Control Register L3 (PBCRL3)	839	Note added Note: * The initial value depends on the operating mode of the LSI.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	PB17MD[2:0]	000	R/W	PB17 Mode Select the function of the PB17/$\overline{CS3}$/IRQ1/TIOC3A pin. 000: PB17 I/O (port) 001: $\overline{CS3}$ output (BSC) 010: Setting prohibited 011: IRQ1 input (INTC) 100: TIOC3A 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	PB16MD[2:0]	000*	R/W	PB16 Mode Select the function of the PB16/$\overline{CS1}$/$\overline{POE1}$/TXD0 pin. • Area 0: 16-bit mode/8-bit mode 000: PB16 I/O (port) 001: $\overline{CS1}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{POE1}$ input (POE2) 101: TXD0 output (SCIF) 110: Setting prohibited 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PB12MD[2:0]	000	R/W	PB12 Mode Select the function of the PB12/$\overline{BREQ}$/TIOC4AS/TXD2 pin. 000: PB12 I/O (port) 001: $\overline{BREQ}$ input (BSC) 010: Setting prohibited 011: Setting prohibited 100: TIOC4AS I/O (MTU2S) 101: TXD2 output (SCIF) 110: Setting prohibited 111: Setting prohibited

Item	Page	Revision (See Manual for Details)
19.1.4 Port B Control Registers H1 to H4, L1 to L4 (PBCRH1 to PBCRH4, PBCRL1 to PBCRL4) (8) Port B Control Register L1 (PBCRL1)	842	Table amended

Bit	Bit Name	Initial Value	R/W	Description
14 to 12	PB3MD[2:0]	000*	R/W	PB3 Mode Select the function of the PB3/CK pin. • Area 0: 16-bit mode/8-bit mode 000: PB3 I/O (port) 001: CK output (CPG) (initial value) 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	PB2MD[2:0]	000*	R/W	PB2 Mode Select the function of the PB2/$\overline{\text{CS0}}$/$\overline{\text{POE4}}$/SCK0 pin. • Area 0: 16-bit mode/8-bit mode 000: PB2 I/O (port) 001: $\overline{\text{CS0}}$ output (BSC) (initial value) 010: Setting prohibited 011: Setting prohibited 100: $\overline{\text{POE4}}$ input (POE2) 101: SCK0 I/O (SCIF) 110: Setting prohibited 111: Setting prohibited

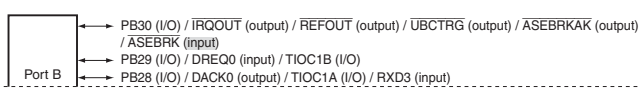
	843	Note added

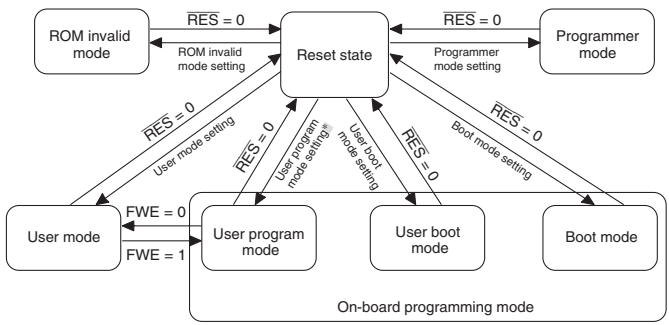
Note: * The initial value depends on the operating mode of the LSI.

19.1.7 Port F Control Register L1 (PFCRL1)	858	Table amended

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	PF0MD[2:0]	000	R/W	PF0 Mode Select the function of the PF0/IRQ0/$\overline{\text{POE7}}$/SCL pin. 000: PF0 input (port) 001: Setting prohibited 010: Setting prohibited 011: IRQ0 input (INTC) 100: $\overline{\text{POE7}}$ input (POE2) 101: SCL I/O (IIC3) 110: Setting prohibited 111: Setting prohibited

20.2 Port B Figure 20.2 Port B	870	Figure amended

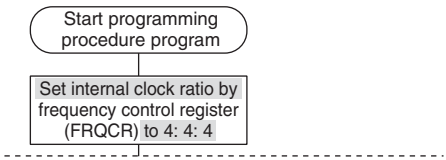
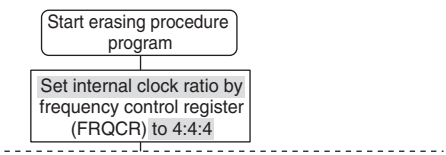


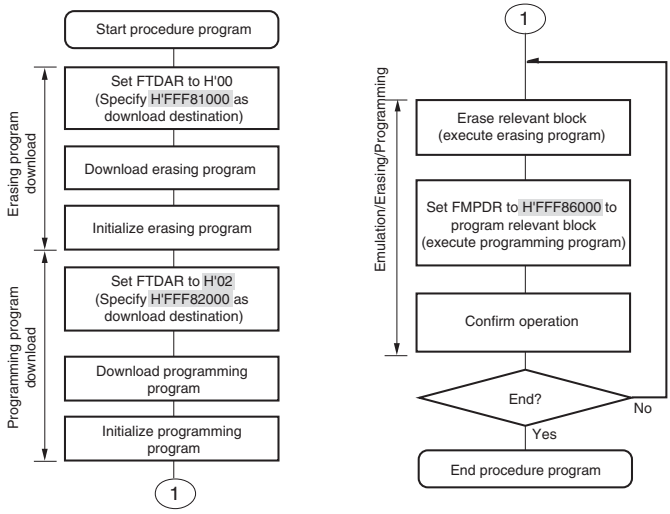
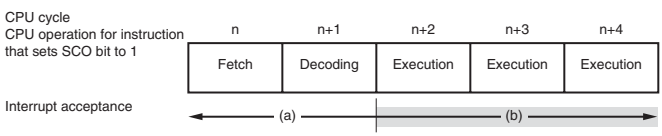
Item	Page	Revision (See Manual for Details)
21.1 Features	886	Description amended • Programming/erasing time The time taken to program 256 bytes of flash memory in a single round is 2 ms (typ.), which is equivalent to 7.8 µs per byte. The erasing time is 80 ms (typ.) per 8-Kbyte block, 600 ms (typ.) per 64-Kbyte block, and 1200 ms (typ.) per 128-Kbyte block. • Number of programming operations The flash memory can be programmed up to 100 times.
21.2.2 Operating Mode Figure 21.2 Mode Transition of Flash Memory	888	Figure amended  Note: * Except for single chip mode.
Table 21.1 Relationship between FWE and MD Pins and Operating Modes	889	Table amended

Pin	Reset State	ROM Invalid Mode	User Mode	User Program Mode	User Boot Mode	Boot Mode	Programmer Mode
RES	0	1	1	1	1	1	Setting value depends on the condition of the specialized PROM programmer.
FWE	0/1	0	0	1	1	1	
MD0	0/1	0/1*1	0/1*2	0	1	0	
MD1	0/1	0	1	1	0	0	

21.4.1 Registers Table 21.5 Register/ Parameter and Target Mode	897	Table amended

		Download	Initialization	Programming	Erasure	Read	RAM Emulation
Programming/ erasing interface registers	FCCS	√	—	—	—	—	—
	FPCS	√	—	—	—	—	—
	FECS	√	—	—	—	—	—
	FKEY	√	—	√	√	—	—
	FMATS	—	—	√*1	√*1	√*2	—
	FTDAR	√	—	—	—	—	—

Item	Page	Revision (See Manual for Details)
21.4.3 Programming/ Erasing Interface Parameters (3) Programming Execution (3.3) Flash Pass/Fail Result Parameter (FPFR: General Register R0 of CPU)	914	Title amended
21.5.2 User Program Mode (2) Programming Procedure in User Program Mode Figure 21.11 Programming Procedure	926	Figure amended  Description amended … Specify 4:4:4 as the frequency division ratios of an internal clock (Iϕ), a bus clock (Bϕ), and a peripheral clock (Pϕ) through the frequency control register (FRQCR).
(3) Erasing Procedure in User Program Mode Figure 21.12 Erasing Procedure	932	Figure amended 
	933	Description amended The frequency division ratio of an internal clock (Iϕ), a bus clock (Bϕ), and a peripheral clock (Pϕ) is specified as 4:4:4 by the frequency control register (FRQCR).

Item	Page	Revision (See Manual for Details)
21.5.2 User Program Mode (4) Erasing and Programming Procedure in User Program Mode Figure 21.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)	934	Figure amended 
	935	Description amended 2. Be sure to initialize both the erasing program and programming program. Initialization by setting the FPEFEQ and FUBRA parameters must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFF81020 in this example) and (download start address for programming program) + 32 bytes (H'FFF82020 in this example).
21.7.2 Interrupts during Programming/Erasing Figure 21.18 Timing of Contention between SCO Download Request and Interrupt Request	945	Figure amended  (a) When the interrupt is accepted at the (n + 1) cycle or before After the interrupt processing completes, the SCO bit is set to 1 and download is executed. (b) When the interrupt is accepted at the (n + 2) cycle or later The interrupt will conflicts with the SCO download request. Ensure that no interrupt is generated.

Item	Page	Revision (See Manual for Details)
21.7.3 Other Notes (2) User Branch Processing Intervals	947	Description amended The intervals for executing the user branch processing differs in programming and erasing. The processing phase also differs. Table 21.10 lists the maximum intervals for initiating the user branch processing when the CPU clock frequency is 40 MHz.
Table 21.10 Initiation Intervals of User Branch Processing		Table amended **Processing Name** / **Maximum Interval** Programming / Approximately 2 ms* Erasing / Approximately 15 ms* Note: * Reference value
		Description amended However, when operation is done with CPU clock of 40 MHz, maximum values of the time until first user branch processing are as shown in table 21.11.
Table 21.11 Initial User Branch Processing Time		Table amended **Processing Name** / **Maximum** Programming / Approximately 2 ms* Erasing / Approximately 15 ms* Note: * Reference value
(7) The Operating Frequency in On-Board Programming Mode	948	Newly added
(8) Programming the User MAT in User Program Mode		Newly added
21.8.1 Specifications of the Standard Serial Communications Interface in Boot Mode (4) Inquiry-and-Selection State (k) New Bit Rate Selection	961	Description amended • Input frequency (2 bytes): The frequency of the clock signal fed to the boot program This should be the frequency in MHz to the second decimal place, multiplied by 100 (for example, if the frequency is 8.882 MHz, the value is truncated to the second decimal place and multiplied by 100, making 888; so H'0378 should be set in this field).
21.8.2 Areas for Storage of the Procedural Program and Data for Programming	979	Description amended 4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been decided.

Item	Page	Revision (See Manual for Details)

22.2.2 RAME and RAMWE Bits
Figure 22.1 Examples of Read/Write before Disabling RAM
	989	Figure amended
		`// For page 1`
		`MOV.L #H'FFF82000,R0`
		`MOV.L @R0,R1`
		`MOV.L R1,@R0`
		`// For page 2`
		`MOV.L #H'FFF84000,R0`
		`MOV.L @R0,R1`
		`MOV.L R1,@R0`
		`// For page 3`
		`MOV.L #H'FFF86000,R0`
		`MOV.L @R0,R1`
		`MOV.L R1,@R0`

24.3.2 Instruction Register (SDIR)	1014	Note added 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.
26.2 Register Bits	1037	Table amended

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	IPR01	IRQ0[3:0]				IRQ1[3:0]			
		IRQ2[3:0]				IRQ3[3:0]			
	IPR02	IRQ4[3:0]				IRQ5[3:0]			
		IRQ6[3:0]				IRQ7[3:0]			
	IPR05	—	—	—	—	—	—	—	—
		ADI[3:0]				—	—	—	—
	IPR06	DMAC0[3:0]				DMAC1[3:0]			
		DMAC2[3:0]				DMAC3[3:0]			
	IPR07	DMAC4[3:0]				DMAC5[3:0]			
		DMAC6[3:0]				DMAC7[3:0]			
	IPR08	CMT0[3:0]				CMT1[3:0]			
		BSC[3:0]				WDT[3:0]			
	IPR09	MTU0(TGI0A to TGI0D)[3:0]				MTU0(TCI0V, TGI0E, TGI0F)[3:0]			
		MTU1(TGI1A, TGI1B)[3:0]				MTU1(TCI1V, TCI1U)[3:0]			

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1038	Table amended

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	IPR10	MTU2(TGI2A, TGI2B)[3:0]				MTU2(TCI2V, TCI2U)[3:0]			
		MTU3(TGI3A to TGI3D)[3:0]				MTU3(TCI3V)[3:0]			
	IPR11	MTU4(TGI4A to TGI4D)[3:0]				MTU4(TCI4V)[3:0]			
		MTU5(TGI5U, TGI5V, TGI5W)[3:0]				POE2(OEI1, OEI2)[3:0]			
	IPR12	MTU3S(TGI3A to TGI3D)[3:0]				MTU3S(TCI3V)[3:0]			
		MTU4S(TGI4A to TGI4D)[3:0]				MTU4S(TCI4V) [3:0]			
	IPR13	MTU5S(TGI5U, TGI5V, TGI5W)[3:0]				POE2(OEI3) [3:0]			
		IIC3[3:0]				—	—	—	—
	IPR14	SCIF0[3:0]				SCIF1[3:0]			
		SCIF2[3:0]				SCIF3[3:0]			
	IPR15	WAVEIF[3:0]				—	—	—	—
		—	—	—	—	—	—	—	—

	1043	Table amended		

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								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1044	Table amended

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_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								
	DAR1								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1045	Table amended

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_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								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1046	Table amended

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_2	—	—	—	—	—	—	—	—
	CHCR_2	TC	—	—	RLD	—	—	—	—
		DO	—	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_2								
	RDAR_2								
	RDMATCR_2	—	—	—	—	—	—	—	—
	SAR_3								
	DAR_3								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1047	Table amended

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_3	—	—	—	—	—	—	—	—
	CHCR_3	TC	—	—	RLD	—	—	—	—
		DO	—	—	—	HE	HIE	AM	AL
		DM[1:0]		SM[1:0]		RS[3:0]			
		DL	DS	TB	TS[1:0]		IE	TE	DE
	RSAR_3								
	RDAR_3								
	RDMATCR_3	—	—	—	—	—	—	—	—
	SAR_4								
	DAR_4								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1048	Table amended

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_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	—	—	—	—	—	—	—	—
	SAR_5								
	DAR_5								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1049	Table amended

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_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								
	DAR_6								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1050	Table amended

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_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								

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1051	Table amended

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	—	—	—	—	—	—	—	—

	1061	Table amended

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	ADDR0	—	—	—	—	ADD0[11:8]			
		ADD0[7:0]							
	ADDR1	—	—	—	—	ADD1[11:8]			
		ADD1[7:0]							
	ADDR2	—	—	—	—	ADD2[11:8]			
		ADD2[7:0]							
	ADDR3	—	—	—	—	ADD3[11:8]			
		ADD3[7:0]							
	ADDR4	—	—	—	—	ADD4[11:8]			
		ADD4[7:0]							
	ADDR5	—	—	—	—	ADD5[11:8]			
		ADD5[7:0]							
	ADDR6	—	—	—	—	ADD6[11:8]			
		ADD6[7:0]							
	ADDR7	—	—	—	—	ADD7[11:8]			
		ADD7[7:0]							

Item	Page	Revision (See Manual for Details)
26.2 Register Bits	1062	Table amended

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	PACRL1	—	PA3MD[2:0]			—	PA2MD[2:0]		
		—	PA1MD[2:0]			—	PA0MD[2:0]		
	PBIORH	—	PB30IOR	PB29IOR	PB28IOR	PB27IOR	PPB26IOR	PB25IOR	PB24IOR
		PB23IOR	PB22IOR	PB21IOR	PB20IOR	PB19IOR	PB18IOR	PB17IOR	PB16IOR

1063 Table amended

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	PBCRL1	—	PB3MD[2:0]			—	PB2MD[2:0]		
		—	PB1MD[2:0]			—	PB0MD[2:0]		
	PDIOR	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	PDCRL1	—	PD3MD[2:0]			—	PD2MD[2:0]		
		—	PD1MD[2:0]			—	PD0MD[2:0]		
	PFCRL1	—	—	—	—	—	—	—	—
		—	PF1MD[2:0]			—	PF0MD[2:0]		

Item	Page	Revision (See Manual for Details)
26.3 Register States in Each Operating Mode	1070	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
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
	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

Item	Page	Revision (See Manual for Details)
26.3 Register States in Each Operating Mode	1071	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	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
	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

Item	Page	Revision (See Manual for Details)
26.3 Register States in Each Operating Mode	1072	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2	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
	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

Item	Page	Revision (See Manual for Details)
26.3 Register States in Each Operating Mode	1073	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU	TDER	Initialized	Retained	Retained	Initialized	Retained
	TSYCR	Initialized	Retained	Retained	Initialized	Retained
	TWCR	Initialized	Retained	Retained	Initialized	Retained
	TOLBR	Initialized	Retained	Retained	Initialized	Retained
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

Item	Page	Revision (See Manual for Details)
26.3 Register States in Each Operating Mode	1074	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TADCOBRA_4S	Initialized	Retained	Retained	Initialized	Retained
	TADCOBRB_4S	Initialized	Retained	Retained	Initialized	Retained
	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

	1075	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
MTU2S	TITCNTS	Initialized	Retained	Retained	Initialized	Retained
	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

Item	Page	Revision (See Manual for Details)
26.3 Register States in Each Operating Mode	1078	Table amended

Module Name	Register Abbreviation	Power-On Reset	Manual Reset	Software Standby	Module Standby	Sleep
PFC	PACRL1	Initialized	Retained	Retained	—	Retained
	PBIORH	Initialized	Retained	Retained	—	Retained
	PBCRL1	Initialized	Retained	Retained	—	Retained
	PDIOR	Initialized	Retained	Retained	—	Retained
	PDCRL1	Initialized	Retained	Retained	—	Retained
	PFCRL1	Initialized	Retained	Retained	—	Retained

27.2 Power-on/Power-off Sequence Table 27.2 Recommended Time for Power-on/Power-off Sequence	1082	Note amended Note: $V_{CC}Q \geq V_{CC} = PLLV_{CC}$ is recommended. Either $V_{CC}Q$, V_{CC}, or $PLLV_{CC}$ power supply can be turned on or off first, though, an undefined period appears until V_{CC} rises to the Min. voltage or after V_{CC} passes the Min. voltage. During these periods, pin or internal states become undefined. Design the system so that such undefined states do not cause a system malfunction. To avoid an increase in the current consumption during the undefined period at power-on, it is recommended that $V_{CC}Q$, V_{CC}, and $PLLV_{CC}$ be turned on simultaneously. This undefined period can be eliminated by turning on the power supplies in the order shown in figure 27.2.
Figure 27.2 Power-on Sequence	1083	Figure added

Item	Page	Revision (See Manual for Details)
27.3 DC Characteristics	1084	Table amended

Table 27.3 DC Characteristics (1) [Common Items]

Item		Symbol	Min.	Typ.	Max.
Current consumption*1	Normal operation	I_{CC}	–	200	310
	Software standby mode	I_{stby}	–	5	60
		PI_{stby}	–	0.1	1
	Sleep mode	I_{sleep}	–	60	100
Input leakage current	All input pins (except PF0, PF1)	$\|I_{in}\|$	–	–	1
	PF0, PF1		–	–	1
Three-state leakage current	Input/output pins, all output pins (off state)	$\|I_{STI}\|$	–	–	1
Input capacitance	All pins	C_{in}	–	–	10

Caution amended

Caution: When neither the A/D converter nor the D/A converter is in use, set $V_{CC} \leq AV_{CC} \leq 5.0\ V \pm 0.5\ V$ and $AV_{SS} = V_{SS}$, and do not leave the AV_{CC}, AV_{SS}, AVREF, and $AVREFV_{SS}$ pins open.

Table 27.3 DC Characteristics (2) [Except for I^2C -Related Pins]	1085	Conditions amended Conditions: $V_{CC} = PLLV_{CC} = 1.4$ to 1.6 V, $V_{CC}Q = 3.0$ to 3.6 V, $V_{SS} = PLLV_{SS} = V_{SS}Q =$ 0 V, Ta = –40°C to +85°C

Item	Page	Revision (See Manual for Details)
27.3 DC Characteristics Table 27.3 DC Characteristics (2) [Except for I^2C -Related Pins]	1086, 1087	Table amended

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	V_{OH}	$V_{CC}Q - 0.8$	–	–	V	I_{OH} = – 5 mA
	All output pins except for above pins		$V_{CC}Q - 0.5$	–	–	V	I_{OH} = – 200 µA
Output low voltage	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	V_{OL}	–	–	0.9	V	I_{OL} = 15 mA
	All output pins except for above pins		–	–	0.4	V	I_{OL} = 2 mA

Table 27.3 DC Characteristics (3) [I^2C-Related Pins*]	1087	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C Table amended

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	V_{IH}	$V_{CC}Q \times 0.7$	–	$V_{CC}Q + 0.3$	V	
Input low voltage	V_{IL}	– 0.3	–	$PV_{CC} \times 0.3$	V	

Table 27.4 Permissible Output Currents	1088	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C

Item	Page	Revision (See Manual for Details)
27.3 DC Characteristics	1088	Table amended

Table 27.4 Permissible Output Currents

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	I_{OL}	–	–	15	mA
	SCL, SDA				10	mA
	Output pins other than above				2	mA
Permissible output low current (total)		ΣI_{OL}	–	–	80	mA
Permissible output high current (per pin)	TIOC3B (PB18), TIOC3D (PB19) TIOC4A to TIOC4D (PB4 to PB7) TIOC3BS (PB21), TIOC3DS (PB20) TIOC4AS to TIOC4DS (PB12, PB13, PB10, PB11)	$-I_{OH}$	–	–	5	mA
	Output pins other than above		–	–	2	mA
Permissible output high current (total)		$\Sigma-I_{OH}$	–	–	25	mA

27.4 AC Characteristics	1089	Conditions amended

Table 27.5 Maximum Operating Frequency

Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C

Table amended

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU (Iϕ)	f	32	–	160	MHz	
	Internal bus, external bus (Bϕ)		32	–	40		
	Peripheral module (Pϕ)		4	–	40		

27.4.1 Clock Timing	1090	Conditions amended

Table 27.6 Clock Timing

Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C

Table amended

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input frequency	f_{EX}	8	10	MHz	Figure 27.3
EXTAL clock input cycle time	t_{EXcyc}	100	125	ns	
EXTAL clock input pulse low width	t_{EXL}	6	–	ns	
EXTAL clock input pulse high width	t_{EXH}	6	–	ns	
EXTAL clock input rise time	t_{EXr}	–	3	ns	
EXTAL clock input fall time	t_{EXf}	–	3	ns	

Item	Page	Revision (See Manual for Details)
27.4.1 Clock Timing Figure 27.3 CKL Clock Input Timing	–	Figure deleted
27.4.2 Control Signal Timing Table 27.7 Control Signal Timing	1093	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C Table amended **Bϕ = 40 MHz** **Item** / **Symbol** / **Min.** / **Max.** / **Unit** MD1, MD0 setup time / t_{MDS} / 20 / – / t_{cyc} Notes amended Notes: 2. In standby mode or when the clock multiplication ratio is changed, t_{RESW} = t_{OSC2} (Min. 10 ms). 3. In standby mode, t_{RESW} = t_{OSC2} (Min. 10 ms).
27.4.3 Bus Timing Table 27.8 Bus Timing	1096	Conditions amended Conditions: Clock mode 6, V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C
	1097	Table amended **Bϕ = 40 MHz*** **Item** / **Symbol** / **Min.** / **Max.** Read data hold time 2 / t_{RDH2} / 2 / –
27.4.4 UBC Trigger Timing Table 27.9 UBC Trigger Timing	1126	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C
27.4.5 DMAC Module Timing Table 27.10 DMAC Module Timing	1126	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C
27.4.6 MTU2, MTU2S Module Timing Table 27.11 MTU2, MTU2S Module Timing	1128	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = –40°C to +85°C

Item	Page	Revision (See Manual for Details)
27.4.7 POE2 Module Timing Table 27.12 POE2 Module Timing	1129	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.8 Watchdog Timer Timing Table 27.13 Watchdog Timer Timing	1129	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.9 SCIF Module Timing Table 27.14 SCIF Module Timing	1130	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.10 IIC3 Module Timing Table 27.15 I^2C Bus Interface 3 Timing	1131	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.11 A/D Trigger Input Timing Table 27.16 A/D Trigger Input Timing	1132	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.12 I/O Port Timing Table 27.17 I/O Port Timing	1133	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.13 H-UDI Related Pin Timing Table 27.18 H-UDI Related Pin Timing	1134	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V, $V_{CC}Q$ = 3.0 to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = 0 V, Ta = −40°C to +85°C
27.4.14 AC Characteristics Measurement Conditions	1136	Description amended • I/O signal reference level: $V_{CC}Q/2$ ($V_{CC}Q$ = 3.0 to 3.6 V, V_{CC} = $PLLV_{CC}$ = 1.4 to 1.6 V)
Figure 27.53 Output Load Circuit		Notes amended Notes: 2. I_{OL} and I_{OH} are shown in table 27.4.
27.6 D/A Converter Characteristics Table 27.20 D/A Converter Characteristics	1138	Conditions amended Conditions: V_{CC} = $PLLV_{CC}$ = 1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = $PLLV_{SS}$ = $V_{SS}Q$ = AV_{SS} = $AVREFV_{SS}$ = 0 V, Ta = −40°C to +85°C

Item	Page	Revision (See Manual for Details)
27.6 D/A Converter Characteristics Table 27.20 D/A Converter Characteristics	1138	Note deleted

Item	Min.	Typ.	Max.	Unit	Test Conditions
Absolute accuracy	–	±2.0	±3.0	LSB	Load resistance 2 MΩ
	–	–	±2.5	LSB	Load resistance 4 MΩ

27.7 Flash Memory Characteristics
Table 27.21 Flash Memory Characteristics

1138

Conditions amended

Conditions: V_{CC} = $PLLV_{CC}$ =1.4 V to 1.6 V, $V_{CC}Q$ = 3.0 V to 3.6 V, V_{SS} = $PLLV_{SS}$ = $V_{CC}Q$ = 0 V, Ta = –40°C to +85°C

Table amended

Item	Symbol	Min.	Typ.	Max.	Unit
Write time *1*2*4	t_P	—	2	20	ms/256 bytes
Erase time *1*2*4	t_E	—	80	260	ms/8-Kbyte block
		—	600	1600	ms/64-Kbyte block
		—	1200	3000	ms/128-Kbyte block
Write time (total) *1*2*4	Σt_P	—	4.5	12	s/512 Kbytes
Erase time (total) *1*2*4	Σt_E	—	4.5	12	s/512 Kbytes
Write and erase time (total) *1*2*4	Σt_{PE}	—	9	24	s/512 Kbytes
Number of rewrite times	N_{WEC}	100*3	—	—	times

Notes amended

Notes: 1. Write time and erase time depend on data.
2. Data transfer time is not included in the write and erase time.
3. Minimum value that guarantees all characteristics after rewriting (guarantees in the range from 1 to Min. value).
4. Characteristics when the number of rewrite times falls within the range including the Min. value.

A. Pin States
Table A.1 Pin States

1139

Table amended

Pin Function		Pin State				
		Reset State				
		Power-On*7				
		Extended without ROM		Extended with ROM	Single chip	Manual
Type	Pin Name	8 Bits	16 Bits			
Clock	CK (clock mode 6)	O			Z	O

Item	Page	Revision (See Manual for Details)
A. Pin States	1142	Table amended

Table A.1 Pin States

Pin Function		Pin State							
		Reset State					Power-Down State		
		Power-On*7							
		Extended without ROM		Extended with ROM	Single chip				
Type	Pin Name	8 Bits	16 Bits			Manual	Software Standby	Sleep	Bus Master-ship Release
Emulator	TDO	O/Z*5				O/Z*5	O/Z*5	O/Z*5	O/Z*5

	1143	Note 7 deleted
B. Product Lineup	1144	Table amended

Table B.1 Product Lineup

Product Type	Product Code	Package
SH7211	R5F72115D160FPV	LQFP2020-144Cu (FP-144LV)

C. Package Dimensions	1145	Figure replaced

Figure C.1 FP-144LV

Index

Numerics

16-bit/32-bit displacement 25

A

A/D conversion time 786
A/D converter (ADC) 769
A/D converter activation 536
A/D converter activation by MTU2 and MTU2S 787
A/D converter start request delaying function 518
Absolute accuracy 791
Absolute address 25
Absolute address accessing 25
Access size and data alignment 229
Access wait control 236
Address errors 87
Address map 179
Address multiplexing 244
Addressing modes 26
Arithmetic operation instructions 44
Auto-refreshing 265
Auto-request mode 329

B

Bank active 258
Banked register and input/output of banks 136
Bit manipulation instructions 52
Bit synchronous circuit 764
Block diagram 7
Boot mode 920
Branch instructions 49
Break detection and processing 722
Break on data access cycle 168
Break on instruction fetch cycle 167
Burst mode 342
Burst read 251
Burst ROM (clock asynchronous) interface 278
Burst ROM (clock synchronous) interface 285
Burst write 255
Bus arbitration 293
Bus state controller (BSC) 175
Bus-released state 54

C

Calculating exception handling vector table addresses 82
Canceling software standby mode (WDT) 658
Cascaded operation 452
Caution on period setting 551
Changing the division ratio 74
Changing the frequency 73, 659
Changing the multiplication rate 73
Clock frequency control circuit 63
Clock operating modes 66
Clock pulse generator (CPG) 61
Clocked synchronous serial format 754
CMCNT count timing 643
Compare match timer (CMT) 637
Complementary PWM mode 472
Conditions for determining number of idle cycles 287
Conflict between byte-write and count-up processes of CMCNT 648
Conflict between word-write and count-up processes of CMCNT 647

Conflict between write and compare-match processes of CMCNT.... 646
Continuous scan mode..... 783
CPU 15
Crystal oscillator..... 63
$\overline{\text{CS}}$n assert period expansion..... 238
Cycle steal mode..... 340

D

D/A converter (DAC)..... 795
D/A output hold function in software standby mode..... 801
Data format in registers 20
Data formats in memory..... 20
Data transfer instructions..... 40
Data transfer with interrupt request signals..... 140
Dead time compensation 529
Deep power-down mode..... 277
Definitions of A/D conversion accuracy..... 791
Delayed branch instructions 23
Direct memory access controller (DMAC) 299
Displacement accessing..... 25
Divider 1..... 63
DMA transfer flowchart 328
DMAC activation 535
DREQ pin sampling timing 345
Dual address mode..... 337

E

Effective address calculation..... 26
Endian..... 229
Equation for getting SCBRR value..... 687
Error protection 941
Exception handling..... 77
Exception handling state..... 54
Exception handling vector table..... 81
Exception source generation immediately after delayed branch instruction..... 96
Exceptions triggered by instructions..... 93
External pulse width measurement 528
External request mode..... 329
External trigger input timing..... 788

F

Fixed mode 333
Flash memory 885
Flash memory configuration..... 891
Full-scale error..... 791

G

General illegal instructions 95
General registers 15
Global base register (GBR)..... 17

H

Hardware protection..... 940
H-UDI commands..... 1014
H-UDI interrupt 114, 1017
H-UDI reset..... 1017

I

I/O ports..... 863
I^2C bus format..... 744
I^2C bus interface 3 (IIC3)..... 725
Immediate data..... 24
Immediate data accessing 24
Immediate data format 21
Initial user branch processing time 947
Initial values of control registers..... 19
Initial values of general registers 19

Initial values of system registers 19
Initiation intervals
of user branch processing 947
Input sampling
and A/D conversion time 785
Instruction features 22
Instruction format 31
Instruction set 35
Integer division instructions 95
Interrupt controller (INTC) 101
Interrupt exception handling 92
Interrupt exception handling vectors
and priorities 118
Interrupt priority level 91
Interrupt response time 129
IRQ interrupts 115

J

Jump table base register (TBR) 17

L

Load-store architecture 22
Logic operation instructions 47
Low-frequency mode 270
Low-power SDRAM 275

M

Manual reset 993
Master receive operation 747
Master transmit operation 745
MCU extension mode 57
MCU operating modes 55
Module standby function 1009
MPX-I/O interface 239
MTU2 functions 352
MTU2 interrupts 534
MTU2 output pin initialization 566
MTU2–MTU2S synchronous
operation 522
MTU2S functions 600
Multi-function timer pulse unit 2
(MTU2) 351
Multi-function timer pulse unit 2S
(MTU2S) 599
Multiplexed pins (port A) 803
Multiplexed pins (port B) 805
Multiplexed pins (port D) 807
Multiplexed pins (port F) 808
Multiply and accumulate register high
(MACH) 18
Multiply and accumulate register low
(MACL) 18
Multiply/Multiply-and-accumulate
operations 23

N

NMI interrupt 114
Noise filter 758
Nonlinearity error 791
Normal space interface 232
Note on bypass capacitor 75
Note on changing operating mode 59
Note on using a PLL oscillation circuit 75
Note on using an external crystal
resonator 75
Notes on board design 793
Notes on noise countermeasures 793

O

Offset error 791
On-board programming mode 920
On-chip peripheral module interrupts 116
On-chip peripheral module request 331
On-chip RAM 987
Operation in asynchronous mode 701

Operation in clocked synchronous mode 711

P

Package 1144
Package dimensions 1145
Page conflict 988
Pin arrangement 8
Pin function controller (PFC) 803
Pin states of this LSI 1139
PLL circuit 1 63
PLL circuit 2 63
POE2 interrupt source 634
Port output enable 2 (POE2) 607
Power-down modes 991
Power-down state 54
Power-on reset 993
Power-on sequence 272
Procedure register (PR) 18
Product code 1144
Program counter (PC) 18
Program execution state 54
Programmer mode 986
PWM Modes 457

Q

Quantization error 791

R

Receive data sampling timing and receive margin (asynchronous mode) 723
Register bank error exception handling 89, 139
Register bank errors 89
Register bank exception 139
Register banks 19, 135
Registers
ACLKCR 72
ADANSR 779
ADCR 773
ADDR0 to ADDR7 780
ADSR 776
ADSTRGR 777
BAMR 147
BAR 146
BBR 148
BRCR 162
CHCR 310
CMCNT 642
CMCOR 642
CMCSR 640
CMNCR 183
CMSTR 639
CSnBCR (n = 0 to 7) 186
CSnWCR (n = 0 to 7) 191
DACR 798
DADR0 797
DADR1 797
DAR 308
DMAOR 321
DMARS0 to DMARS3 325
DMATCR 309
DPFR 906
FCCS 898
FEBS 917
FECS 901
FKEY 902
FMATS 903
FMPAR 912
FMPDR 913
FPCS 901
FPEFEQ 908
FPFR 911, 914, 918
FRQCR 68
FTDAR 904
FUBRA 909

IBCR 111
IBNR 112
ICCR1 729
ICCR2 732
ICDRR 742
ICDRS 742
ICDRT 741
ICIER 736
ICMR 734
ICR0 107
ICR1 108
ICSR 738
ICSR1 613
ICSR2 618
ICSR3 622
IFCR 859
IPR01, IPR02, IPR05 to IPR15 105
IRQRR 109
MCLKCR 71
NF2CYC 743
OCSR1 617
OCSR2 621
PACRH1 817
PACRH2 814
PACRH3 812
PACRL1 825
PACRL2 823
PACRL3 821
PACRL4 819
PADRH 865
PADRL 866
PAIORH 811
PAIORL 811
PAPRH 868
PAPRL 869
PBCRH1 834
PBCRH2 832
PBCRH3 830
PBCRH4 828
PBCRL1 842
PBCRL2 840
PBCRL3 838
PBCRL4 836
PBDRH 872
PBDRL 873
PBIORH 827
PBIORL 827
PBPRH 875
PBPRL 876
PDCRL1 855
PDCRL2 853
PDCRL3 849
PDCRL4 845
PDDRL 878
PDIOR 844
PDPRL 880
PFCRL1 857
PFDR 882
POECR1 626
POECR2 627
RDAR 319
RDMATCR 320
RSAR 318
RTCNT 227
RTCOR 228
RTCSR 225
SAR (DMAC) 307
SAR (IIC3) 741
SCBRR 687
SCFCR 692
SCFDR 694
SCFRDR 670
SCFSR 679
SCFTDR 671
SCLSR 696
SCRSR 670
SCSCR 675
SCSEMR 698
SCSMR 672
SCSPTR 695

SCTSR 671
SDBPR 1013
SDCR 221
SDIR 1013
SPOER 624
STBCR 995
STBCR2 996
STBCR3 997
STBCR4 999
SYSCR1 1001
SYSCR2 1003
TADCOBRA_4 409
TADCOBRB_4 409
TADCORA_4 409
TADCORB_4 409
TADCR 406
TBTER 434
TBTM 401
TCBR 431
TCDR 430
TCNT 410
TCNTCMPCLR 388
TCNTS 429
TCR 362
TCSYSTR 415
TDDR 430
TDER 436
TGCR 427
TGR 410
TICCR 403
TIER 389
TIOR 369
TITCNT 433
TITCR 431
TMDR 366
TOCR1 420
TOCR2 423
TOER 419
TOLBR 426
TRWER 418
TSR 394
TSTR 411
TSYCR 404
TSYR 413
TWCR 437
WAVECR1 861
WAVECR2 860
WRCSR 655
WTCNT 652
WTCSR 653
Relationship between access size and number of bursts 251
Relationship between clock operating mode and frequency range 67
Relationship between refresh requests and bus cycles 269
Reset state 54
Reset-synchronized PWM mode 469
Restoration from bank 137
Restoration from stack 138
Restriction on DMAC usage 722
RISC-type instruction set 22
Round-robin mode 333

S

Saving to bank 136
Saving to stack 138
SCIF interrupt sources 720
SDRAM interface 243
Self-refreshing 267
Sending a break signal 722
Serial communication interface with FIFO (SCIF) 665
Setting analog input voltage 801
Shift instructions 48
Sign extension of word data 22
Single address mode 339
Single chip mode 57
Single read 254

Single write 257
Single-cycle scan mode 781
Slave receive operation 752
Slave transmit operation 749
Sleep mode 1005
Slot illegal instructions 94
Software protection 941
Software standby mode 1005
SRAM interface with byte selection 280
Stack after interrupt exception handling 128
Stack status after exception handling ends 97
Standby control circuit 64
Status register (SR) 16
Supported DMA transfers 336
System control instructions 50

T

T bit 23
TAP controller 1015
TDO output timing 1016
The address map for the operating modes 58
Timing to clear an interrupt source 141
Transfer rate 731
Trap instructions 94
Types of exception handling and priority order 77

U

Unconditional branch instructions with no delay slot 23
User boot mode 935
User break controller (UBC) 143
User break interrupt 114
User debugging interface (H-UDI) 1011
User MAT 892
User program mode 924
Using interval timer mode 662
Using watchdog timer mode 660

V

Vector base register (VBR) 17

W

Wait between access cycles 286
Watchdog timer (WDT) 649
WAVE Interface (WAVEIF) 1019

Renesas 32-Bit RISC Microcomputer
Hardware Manual
SH7211 Group

Publication Date:	Rev.1.00, October 30, 2006
	Rev.2.00, May 8, 2008
Published by:	Sales Strategic Planning Div.
	Renesas Technology Corp.
Edited by:	Customer Support Department
	Global Strategic Communication Div.
	Renesas Solutions Corp.

© 2008. Renesas Technology Corp., All rights reserved. Printed in Japan.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology 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: <603> 7955-9390, Fax: <603> 7955-9510

SH7211 Group
Hardware Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan