User's Manual

32

SH7201 Group

User's Manual: Hardware

Renesas 32-Bit RISC Microcomputer
SuperH™ RISC engine Family / SH7200 Series

R5S72011

Renesas Electronics
www.renesas.com

Rev.3.00 Sep 2010

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

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

2. Processing at Power-on

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

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

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

Configuration of This Manual

This manual comprises the following items:

1. General Precautions in the Handling of MPU/MCU Products
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 for 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 that includes a Renesas 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.

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 28, List of Registers.

- Description of Numbers and Symbols

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

(1) Overall notation

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

(2) Register notation

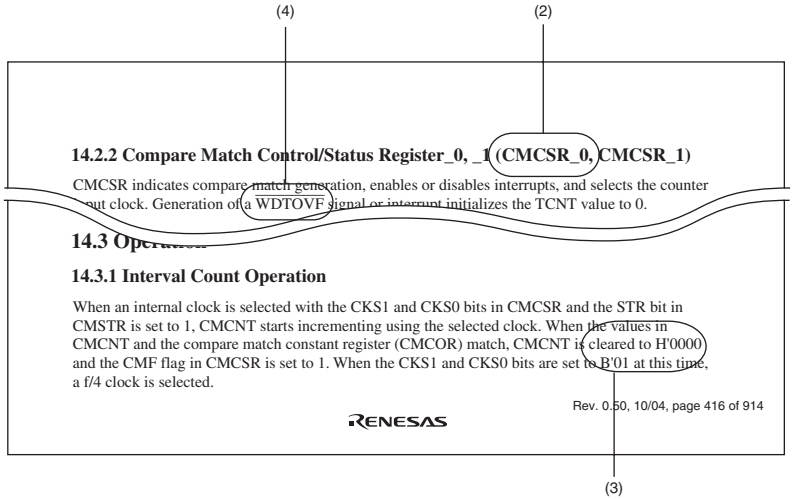
The style "register name"_"instance number" is used in cases where there is more than one instance of the same function or similar functions.
[Example] CMCSR_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

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

(4) Notation for active-low

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

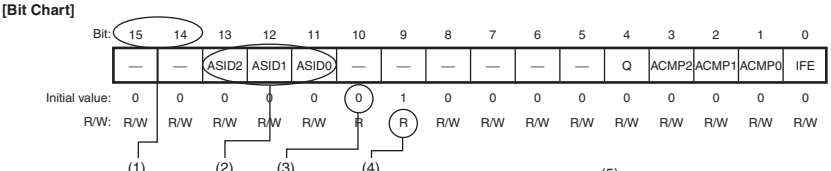


Note: The bit names and sentences in the above figure are examples and do not refer to specific data in this manual.

- Description of Registers

  Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

[Bit Chart]



[Table of Bits]

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

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

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

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

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

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

(4) R/W
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.
The notation is as follows:
R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable.
However, writing is only performed to flag clearing.
R: The bit or field is readable.
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.
W: The bit or field is writable.

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

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

Contents

Section 1 Overview 1
1.1 SH7201 Group Features 1
1.2 Product Lineup 8
1.3 Block Diagram 9
1.4 Pin Assignments 10
1.5 Pin Functions 11

Section 2 CPU 19
2.1 Register Configuration 19
2.1.1 General Registers 19
2.1.2 Control Registers 20
2.1.3 System Registers 22
2.1.4 Register Banks 23
2.1.5 Initial Values of Registers 23
2.2 Data Formats 24
2.2.1 Data Format in Registers 24
2.2.2 Data Formats in Memory 24
2.2.3 Immediate Data Format 25
2.3 Instruction Features 26
2.3.1 RISC-Type Instruction Set 26
2.3.2 Addressing Modes 30
2.3.3 Instruction Format 34
2.4 Instruction Set 38
2.4.1 Instruction Set by Classification 38
2.4.2 Data Transfer Instructions 44
2.4.3 Arithmetic Operation Instructions 48
2.4.4 Logic Operation Instructions 51
2.4.5 Shift Instructions 52
2.4.6 Branch Instructions 53
2.4.7 System Control Instructions 54
2.4.8 Floating Point Operation Instructions 56
2.4.9 FPU-Related CPU Instructions 58
2.4.10 Bit Manipulation Instructions 59
2.5 Processing States 60

Section 3 Floating-Point Unit (FPU) 63
3.1 Features 63
3.2 Data Formats 63
3.2.1 Floating-Point Format 63
3.2.2 Non-Numbers (NaN) 65
3.2.3 Denormalized Numbers 66
3.3 Register Descriptions 67
3.3.1 Floating-Point Registers 67
3.3.2 Floating-Point Status/Control Register (FPSCR) 68
3.3.3 Floating-Point Communication Register (FPUL) 69
3.4 Rounding 70
3.5 FPU Exceptions 71
3.5.1 FPU Exception Sources 71
3.5.2 FPU Exception Handling 71

Section 4 Clock Pulse Generator (CPG) 73
4.1 Features 73
4.2 Input/Output Pins 76
4.3 Clock Operating Modes 77
4.4 Register Descriptions 83
4.4.1 Frequency Control Register (FRQCR) 83
4.4.2 CKIO Control Register (CKIOCR) 86
4.5 Changing the Frequency 87
4.5.1 Changing the Multiplication Rate 87
4.5.2 Changing the Division Ratio 88
4.6 Notes on Board Design 89
4.6.1 Note on Inputting External Clock 89
4.6.2 Note on Using Crystal Resonator 89
4.6.3 Note on Resonator 90
4.6.4 Note on Using a PLL Oscillation Circuit 90
4.6.5 Note on Changing the Multiplication Rate 90

Section 5 Exception Handling 91
5.1 Overview 91
5.1.1 Types of Exception Handling and Priority 91
5.1.2 Exception Handling Operations 93
5.1.3 Exception Handling Vector Table 95
5.2 Resets 97
5.2.1 Input/Output Pins 97

5.2.2 Types of Reset 97
5.2.3 Power-On Reset 98
5.2.4 Manual Reset 100
5.3 Address Errors 101
5.3.1 Address Error Sources 101
5.3.2 Address Error Exception Handling 102
5.4 Bus Error 103
5.4.1 Bus Error Generation Source 103
5.4.2 Bus Error Exception Handling 103
5.5 Register Bank Errors 104
5.5.1 Register Bank Error Sources 104
5.5.2 Register Bank Error Exception Handling 104
5.6 Interrupts 105
5.6.1 Interrupt Sources 105
5.6.2 Interrupt Priority Level 106
5.6.3 Interrupt Exception Handling 107
5.7 Exceptions Triggered by Instructions 108
5.7.1 Types of Exceptions Triggered by Instructions 108
5.7.2 Trap Instructions 109
5.7.3 Slot Illegal Instructions 109
5.7.4 General Illegal Instructions 109
5.7.5 Integer Division Exceptions 110
5.7.6 FPU Exceptions 110
5.8 When Exception Sources Are Not Accepted 111
5.9 Stack Status after Exception Handling Ends 112
5.10 Usage Notes 114
5.10.1 Value of Stack Pointer (SP) 114
5.10.2 Value of Vector Base Register (VBR) 114
5.10.3 Address Errors Caused by Stacking of Address Error Exception Handling 114

Section 6 Interrupt Controller (INTC) 115
6.1 Features 115
6.2 Input/Output Pins 117
6.3 Register Descriptions 117
6.3.1 Interrupt Priority Registers 01, 02, 05 to 16 (IPR01, IPR02, IPR05 to IPR16) 119
6.3.2 Interrupt Control Register 0 (ICR0) 121
6.3.3 Interrupt Control Register 1 (ICR1) 122
6.3.4 Interrupt Control Register 2 (ICR2) 123
6.3.5 IRQ Interrupt Request Register (IRQRR) 123

6.3.6 PINT Interrupt Enable Register (PINTER) ... 125
6.3.7 PINT Interrupt Request Register (PIRR) ... 126
6.3.8 Bank Control Register (IBCR) ... 127
6.3.9 Bank Number Register (IBNR) ... 128
6.3.10 DMA Transfer Request Enable Register 0 (DREQER0) ... 129
6.3.11 DMA Transfer Request Enable Register 1 (DREQER1) ... 130
6.3.12 DMA Transfer Request Enable Register 2 (DREQER2) ... 131
6.3.13 DMA Transfer Request Enable Register 3 (DREQER3) ... 132
6.4 Interrupt Sources ... 133
6.4.1 NMI Interrupt ... 133
6.4.2 User Break Interrupt ... 133
6.4.3 H-UDI Interrupt ... 133
6.4.4 IRQ Interrupts ... 134
6.4.5 PINT Interrupts ... 135
6.4.6 On-Chip Peripheral Module Interrupts ... 135
6.5 Interrupt Exception Handling Vector Table and Priority ... 136
6.6 Operation ... 146
6.6.1 Interrupt Operation Sequence ... 146
6.6.2 Stack after Interrupt Exception Handling ... 148
6.7 Interrupt Response Time ... 149
6.8 Register Banks ... 154
6.8.1 Register Banks and Bank Control Registers ... 155
6.8.2 Bank Save and Restore Operations ... 155
6.8.3 Save and Restore Operations after Saving to All Banks ... 157
6.8.4 Register Bank Exception ... 158
6.8.5 Register Bank Error Exception Handling ... 158
6.9 Data Transfer with Interrupt Request Signals ... 159
6.9.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but not DMAC Activation ... 159
6.9.2 Handling Interrupt Request Signals as Sources for DMAC Activation but not CPU Interrupt ... 159
6.10 Usage Note ... 160
6.10.1 Timing to Clear an Interrupt Source ... 160

Section 7 User Break Controller (UBC) ... 161
7.1 Features ... 161
7.2 Input/Output Pin ... 163
7.3 Register Descriptions ... 163
7.3.1 Break Address Register (BAR) ... 164
7.3.2 Break Address Mask Register (BAMR) ... 165

7.3.3 Break Data Register (BDR) ... 166
7.3.4 Break Data Mask Register (BDMR) ... 167
7.3.5 Break Bus Cycle Register (BBR) ... 168
7.3.6 Break Control Register (BRCR) ... 170
7.4 Operation ... 173
7.4.1 Flow of the User Break Operation ... 173
7.4.2 Break on Instruction Fetch Cycle ... 174
7.4.3 Break on Data Access Cycle ... 175
7.4.4 Value of Saved Program Counter ... 176
7.4.5 Usage Examples ... 177
7.5 Usage Notes ... 180

Section 8 Cache ... 181
8.1 Features ... 181
8.1.1 Cache Structure ... 181
8.2 Register Descriptions ... 184
8.2.1 Cache Control Register 1 (CCR1) ... 184
8.2.2 Cache Control Register 2 (CCR2) ... 186
8.3 Operation ... 189
8.3.1 Searching Cache ... 189
8.3.2 Read Access ... 191
8.3.3 Prefetch Operation (Only for Operand Cache) ... 191
8.3.4 Write Operation (Only for Operand Cache) ... 191
8.3.5 Write-Back Buffer (Only for Operand Cache) ... 192
8.3.6 Coherency of Cache and External Memory ... 194
8.4 Memory-Mapped Cache ... 194
8.4.1 Address Array ... 194
8.4.2 Data Array ... 195
8.4.3 Usage Examples ... 197
8.4.4 Notes ... 198

Section 9 Bus State Controller (BSC) ... 199
9.1 Features ... 199
9.2 Input/Output Pins ... 201
9.3 Area Overview ... 203
9.3.1 Address Map ... 203
9.3.2 Data Bus Width and Pin Function Setting for Individual Areas ... 204
9.4 Register Descriptions ... 205
9.4.1 CSn Control Register (CSnCNT) (n = 0 to 6) ... 207
9.4.2 CSn Recovery Cycle Setting Register (CSnREC) (n = 0 to 6) ... 209

9.4.3 SDRAMCm Control Register (SDCmCNT) (m = 0, 1) ... 211
9.4.4 CSn Mode Register (CSMODn) (n = 0 to 6) ... 212
9.4.5 CSn Wait Control Register 1 (CS1WCNTn) (n = 0 to 6) ... 215
9.4.6 CSn Wait Control Register 2 (CS2WCNTn) (n = 0 to 6) ... 217
9.4.7 SDRAM Refresh Control Register 0 (SDRFCNT0) ... 220
9.4.8 SDRAM Refresh Control Register 1 (SDRFCNT1) ... 221
9.4.9 SDRAM Initialization Register 0 (SDIR0) ... 223
9.4.10 SDRAM Initialization Register 1 (SDIR1) ... 225
9.4.11 SDRAM Power-Down Control Register (SDPWDCNT) ... 226
9.4.12 SDRAM Deep-Power-Down Control Register (SDDPWDCNT) ... 227
9.4.13 SDRAMm Address Register (SDmADR) (m = 0, 1) ... 228
9.4.14 SDRAMm Timing Register (SDmTR) (m = 0, 1) ... 229
9.4.15 SDRAMm Mode Register (SDmMOD) (m = 0, 1) ... 231
9.4.16 SDRAM Status Register (SDSTR) ... 232
9.4.17 SDRAM Clock Stop Control Signal Setting Register (SDCKSCNT) ... 234
9.4.18 AC Characteristics Switching Register (ACSWR) ... 236
9.5 Operation ... 237
9.5.1 CSC Interface ... 237
9.5.2 SDRAM Interface ... 247
9.6 Usage Note ... 283
9.6.1 Note on Power-on Reset Exception Handling and Deep Standby Mode Cancellation ... 283
9.6.2 Write Buffer ... 283
9.6.3 Note on Transition to Software Standby Mode or Deep Standby Mode ... 283

Section 10 Bus Monitor ... 285
10.1 Register Descriptions ... 285
10.1.1 Bus Monitor Enable Register (SYCBEEN) ... 286
10.1.2 Bus Monitor Status Register 1 (SYCBESTS1) ... 287
10.1.3 Bus Monitor Status Register 2 (SYCBESTS2) ... 289
10.1.4 Bus Error Control Register (SYCBESW) ... 292
10.2 Bus Monitor Function ... 293
10.2.1 Operation when a Bus Error is Detected ... 293
10.2.2 Illegal Address Access Detection Function ... 294
10.2.3 Bus Timeout Detection Function ... 296
10.2.4 Combinations of Masters and Bus Errors ... 297
10.3 Usage Note ... 298
10.3.1 Operation when the CPU is Not Notified of a Bus Error ... 298

Section 11 Direct Memory Access Controller (DMAC) 299
11.1 Features 299
11.2 Input/Output Pins 301
11.3 Register Descriptions 302
11.3.1 DMA Current Source Address Register (DMCSADR) 306
11.3.2 DMA Current Destination Address Register (DMCDADR) 307
11.3.3 DMA Current Byte Count Register (DMCBCT) 308
11.3.4 DMA Reload Source Address Register (DMRSADR) 309
11.3.5 DMA Reload Destination Address Register (DMRDADR) 310
11.3.6 DMA Reload Byte Count Register (DMRBCT) 311
11.3.7 DMA Mode Register (DMMOD) 312
11.3.8 DMA Control Register A (DMCNTA) 318
11.3.9 DMA Control Register B (DMCNTB) 326
11.3.10 DMA Activation Control Register (DMSCNT) 332
11.3.11 DMA Interrupt Control Register (DMICNT) 333
11.3.12 DMA Common Interrupt Control Register (DMICNTA) 334
11.3.13 DMA Interrupt Status Register (DMISTS) 335
11.3.14 DMA Transfer End Detection Register (DMEDET) 336
11.3.15 DMA Arbitration Status Register (DMASTS) 338
11.4 Operation 340
11.4.1 DMA Transfer Mode 340
11.4.2 DMA Transfer Condition 342
11.4.3 DMA Activation 346
11.5 Completion of DMA Transfer and Interrupts 347
11.5.1 Completion of DMA Transfer 347
11.5.2 DMA Interrupt Requests 348
11.5.3 DMA End Signal Output 350
11.6 Suspending, Restarting, and Stopping of DMA Transfer 352
11.6.1 Suspending and Restarting DMA Transfer 352
11.6.2 Stopping DMA Transfer on Any Channel 352
11.7 DMA Requests 353
11.7.1 Sources of DMA Requests 353
11.7.2 Synchronous Circuits for DMA Request Signals 353
11.7.3 Sense Mode for DMA Requests 354
11.8 Determining DMA Channel Priority 357
11.8.1 Channel Priority Order 357
11.8.2 Operation during Multiple DMA Requests 357
11.8.3 Output of the DMA Acknowledge and DNA Active Signals 358
11.9 Units of Transfer and Positioning of Bytes for Transfer 360

11.10 Reload Function 361
11.11 Rotate Function 363
11.12 Transfer Speed 364
11.13 Usage Note 366
11.13.1 Note on Making a Transition To Software Standby Mode or Deep Standby Mode 366

Section 12 Multi-Function Timer Pulse Unit 2 (MTU2) 367
12.1 Features 367
12.2 Input/Output Pins 373
12.3 Register Descriptions 374
12.3.1 Timer Control Register (TCR) 380
12.3.2 Timer Mode Register (TMDR) 384
12.3.3 Timer I/O Control Register (TIOR) 387
12.3.4 Timer Compare Match Clear Register (TCNTCMPCLR) 406
12.3.5 Timer Interrupt Enable Register (TIER) 407
12.3.6 Timer Status Register (TSR) 412
12.3.7 Timer Buffer Operation Transfer Mode Register (TBTM) 419
12.3.8 Timer Input Capture Control Register (TICCR) 420
12.3.9 Timer A/D Converter Start Request Control Register (TADCR) 421
12.3.10 Timer A/D Converter Start Request Cycle Set Registers (TADCORA_4 and TADCORB_4) 424
12.3.11 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA_4 and TADCOBRB_4) 425
12.3.12 Timer Counter (TCNT) 425
12.3.13 Timer General Register (TGR) 426
12.3.14 Timer Start Register (TSTR) 427
12.3.15 Timer Synchronous Register (TSYR) 429
12.3.16 Timer Counter Synchronous Start Register (TCSYSTR) 431
12.3.17 Timer Read/Write Enable Register (TRWER) 433
12.3.18 Timer Output Master Enable Register (TOER) 434
12.3.19 Timer Output Control Register 1 (TOCR1) 435
12.3.20 Timer Output Control Register 2 (TOCR2) 438
12.3.21 Timer Output Level Buffer Register (TOLBR) 441
12.3.22 Timer Gate Control Register (TGCR) 442
12.3.23 Timer Subcounter (TCNTS) 444
12.3.24 Timer Dead Time Data Register (TDDR) 445
12.3.25 Timer Cycle Data Register (TCDR) 445
12.3.26 Timer Cycle Buffer Register (TCBR) 446
12.3.27 Timer Interrupt Skipping Set Register (TITCR) 446

12.3.28 Timer Interrupt Skipping Counter (TITCNT) ... 448
12.3.29 Timer Buffer Transfer Set Register (TBTER) ... 449
12.3.30 Timer Dead Time Enable Register (TDER) ... 451
12.3.31 Timer Waveform Control Register (TWCR) ... 452
12.3.32 Bus Master Interface ... 453
12.4 Operation ... 454
12.4.1 Basic Functions ... 454
12.4.2 Synchronous Operation ... 460
12.4.3 Buffer Operation ... 462
12.4.4 Cascaded Operation ... 467
12.4.5 PWM Modes ... 472
12.4.6 Phase Counting Mode ... 477
12.4.7 Reset-Synchronized PWM Mode ... 484
12.4.8 Complementary PWM Mode ... 487
12.4.9 A/D Converter Start Request Delaying Function ... 524
12.4.10 External Pulse Width Measurement ... 528
12.4.11 Dead Time Compensation ... 529
12.4.12 TCNT Capture at Crest and/or Trough in Complementary PWM Operation ... 531
12.5 Interrupt Sources ... 532
12.5.1 Interrupt Sources and Priorities ... 532
12.5.2 DMAC Activation ... 534
12.5.3 A/D Converter Activation ... 534
12.6 Operation Timing ... 536
12.6.1 Input/Output Timing ... 536
12.6.2 Interrupt Signal Timing ... 543
12.7 Usage Notes ... 548
12.7.1 Module Standby Mode Setting ... 548
12.7.2 Input Clock Restrictions ... 548
12.7.3 Caution on Period Setting ... 549
12.7.4 Contention between TCNT Write and Clear Operations ... 549
12.7.5 Contention between TCNT Write and Increment Operations ... 550
12.7.6 Contention between TGR Write and Compare Match ... 551
12.7.7 Contention between Buffer Register Write and Compare Match ... 552
12.7.8 Contention between Buffer Register Write and TCNT Clear ... 553
12.7.9 Contention between TGR Read and Input Capture ... 554
12.7.10 Contention between TGR Write and Input Capture ... 555
12.7.11 Contention between Buffer Register Write and Input Capture ... 556
12.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connection ... 556
12.7.13 Counter Value during Complementary PWM Mode Stop ... 558

12.7.14 Buffer Operation Setting in Complementary PWM Mode 558
12.7.15 Reset Sync PWM Mode Buffer Operation and Compare Match Flag 559
12.7.16 Overflow Flags in Reset Synchronous PWM Mode 560
12.7.17 Contention between Overflow/Underflow and Counter Clearing 561
12.7.18 Contention between TCNT Write and Overflow/Underflow 562
12.7.19 Cautions on Transition from Normal Operation or PWM Mode 1 to Reset-Synchronized PWM Mode 562
12.7.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 563
12.7.21 Interrupts in Module Standby Mode 563
12.7.22 Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection 563
12.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode 564
12.8 MTU2 Output Pin Initialization 566
12.8.1 Operating Modes 566
12.8.2 Reset Start Operation 566
12.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc. 567
12.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc. 568

Section 13 8-Bit Timers (TMR) 599
13.1 Features 599
13.2 Input/Output Pins 601
13.3 Register Descriptions 601
13.3.1 Timer Counter (TCNT) 602
13.3.2 Time Constant Register A (TCORA) 602
13.3.3 Time Constant Register B (TCORB) 603
13.3.4 Timer Control Register (TCR) 603
13.3.5 Timer Counter Control Register (TCCR) 605
13.3.6 Timer Control/Status Register (TCSR) 607
13.4 Operation 611
13.4.1 Pulse Output 611
13.4.2 Reset Input 612
13.5 Operation Timing 613
13.5.1 TCNT Count Timing 613
13.5.2 Timing of CMFA and CMFB Setting at Compare Match 614
13.5.3 Timing of Timer Output at Compare Match 614
13.5.4 Timing of Counter Clear by Compare Match 615
13.5.5 Timing of TCNT External Reset 615
13.5.6 Timing of Overflow Flag (OVF) Setting 616

13.6 Operation with Cascaded Connection 617
13.6.1 16-Bit Counter Mode 617
13.6.2 Compare Match Count Mode 617
13.7 Interrupt Sources 618
13.7.1 Interrupt Sources 618
13.7.2 A/D Converter Activation 618
13.8 Usage Notes 619
13.8.1 Notes on Setting Cycle 619
13.8.2 Conflict between TCNT Write and Clear 619
13.8.3 Conflict between TCNT Write and Increment 620
13.8.4 Conflict between TCOR Write and Compare Match 620
13.8.5 Conflict between Compare Matches A and B 621
13.8.6 Switching of Internal Clocks and TCNT Operation 621
13.8.7 Mode Setting with Cascaded Connection 623
13.8.8 Module Standby Setting 623
13.8.9 Interrupts in Module Standby Mode 623

Section 14 Watchdog Timer (WDT) 625
14.1 Features 625
14.2 Input/Output Pin 626
14.3 Register Descriptions 627
14.3.1 Watchdog Timer Counter (WTCNT) 627
14.3.2 Watchdog Timer Control/Status Register (WTCSR) 628
14.3.3 Watchdog Reset Control/Status Register (WRCSR) 630
14.3.4 Notes on Register Access 631
14.4 WDT Usage 633
14.4.1 Canceling Software Standby Mode 633
14.4.2 Changing the Frequency 633
14.4.3 Using Watchdog Timer Mode 634
14.4.4 Using Interval Timer Mode 635
14.5 Usage Notes 636
14.5.1 Timer Variation 636
14.5.2 Prohibition against Setting H'FF to WTCNT 636
14.5.3 Interval Timer Overflow Flag 636
14.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal 637
14.5.5 Manual Reset in Watchdog Timer Mode 637

Section 15 Realtime Clock (RTC) 639
15.1 Features 639
15.2 Input/Output Pin 641

15.3 Register Descriptions 641
15.3.1 64-Hz Counter (R64CNT) 642
15.3.2 Second Counter (RSECCNT) 643
15.3.3 Minute Counter (RMINCNT) 644
15.3.4 Hour Counter (RHRCNT) 645
15.3.5 Day of Week Counter (RWKCNT) 646
15.3.6 Date Counter (RDAYCNT) 647
15.3.7 Month Counter (RMONCNT) 648
15.3.8 Year Counter (RYRCNT) 649
15.3.9 Second Alarm Register (RSECAR) 650
15.3.10 Minute Alarm Register (RMINAR) 651
15.3.11 Hour Alarm Register (RHRAR) 652
15.3.12 Day of Week Alarm Register (RWKAR) 653
15.3.13 Date Alarm Register (RDAYAR) 654
15.3.14 Month Alarm Register (RMONAR) 655
15.3.15 Year Alarm Register (RYRAR) 656
15.3.16 RTC Control Register 1 (RCR1) 657
15.3.17 RTC Control Register 2 (RCR2) 659
15.3.18 RTC Control Register 3 (RCR3) 661
15.4 Operation 662
15.4.1 Initial Settings of Registers after Power-On 662
15.4.2 Setting Time 662
15.4.3 Reading Time 663
15.4.4 Alarm Function 664
15.5 Usage Notes 665
15.5.1 Register Writing during RTC Count 665
15.5.2 Use of Realtime Clock (RTC) Periodic Interrupts 665
15.5.3 Transition to Standby Mode after Setting Register 665
15.5.4 Crystal Oscillator Circuit for RTC 666
15.5.5 Procedure for Setting the 30-Second Adjustment Function 667

Section 16 Serial Communication Interface with FIFO (SCIF) 669
16.1 Features 669
16.2 Input/Output Pins 671
16.3 Register Descriptions 671
16.3.1 Receive Shift Register (SCRSR) 675
16.3.2 Receive FIFO Data Register (SCFRDR) 675
16.3.3 Transmit Shift Register (SCTSR) 676
16.3.4 Transmit FIFO Data Register (SCFTDR) 676
16.3.5 Serial Mode Register (SCSMR) 677

16.3.6 Serial Control Register (SCSCR) ... 680
16.3.7 Serial Status Register (SCFSR) ... 684
16.3.8 Bit Rate Register (SCBRR) ... 692
16.3.9 FIFO Control Register (SCFCR) ... 700
16.3.10 FIFO Data Count Register (SCFDR) ... 702
16.3.11 Serial Port Register (SCSPTR) ... 703
16.3.12 Line Status Register (SCLSR) ... 705
16.4 Operation ... 706
16.4.1 Overview ... 706
16.4.2 Operation in Asynchronous Mode ... 708
16.4.3 Operation in Clocked Synchronous Mode ... 717
16.5 SCIF Interrupts ... 725
16.6 Usage Notes ... 726
16.6.1 SCFTDR Writing and TDFE Flag ... 726
16.6.2 SCFRDR Reading and RDF Flag ... 726
16.6.3 Restriction on DMAC Usage ... 727
16.6.4 Break Detection and Processing ... 727
16.6.5 Sending a Break Signal ... 727
16.6.6 Receive Data Sampling Timing and Receive Margin (Asynchronous Mode) ... 728

Section 17 I^2C Bus Interface 3 (IIC3) ... 729
17.1 Features ... 729
17.2 Input/Output Pins ... 731
17.3 Register Descriptions ... 732
17.3.1 I^2C Bus Control Register 1 (ICCR1) ... 733
17.3.2 I^2C Bus Control Register 2 (ICCR2) ... 736
17.3.3 I^2C Bus Mode Register (ICMR) ... 738
17.3.4 I^2C Bus Interrupt Enable Register (ICIER) ... 740
17.3.5 I^2C Bus Status Register (ICSR) ... 742
17.3.6 Slave Address Register (SAR) ... 745
17.3.7 I^2C Bus Transmit Data Register (ICDRT) ... 746
17.3.8 I^2C Bus Receive Data Register (ICDRR) ... 746
17.3.9 I^2C Bus Shift Register (ICDRS) ... 746
17.3.10 NF2CYC Register (NF2CYC) ... 747
17.4 Operation ... 748
17.4.1 I^2C Bus Format ... 748
17.4.2 Master Transmit Operation ... 749
17.4.3 Master Receive Operation ... 751
17.4.4 Slave Transmit Operation ... 753
17.4.5 Slave Receive Operation ... 756

17.4.6 Clocked Synchronous Serial Format ... 757
17.4.7 Noise Filter ... 761
17.4.8 Example of Use ... 762
17.5 Interrupt Requests ... 766
17.6 Bit Synchronous Circuit ... 767
17.7 Usage Note ... 770
17.7.1 Issuance of Stop Condition and Start Condition (Retransmission) ... 770
17.7.2 Note on Setting for Multi-Master Operation ... 770
17.7.3 Note on Master Receive Mode ... 770
17.7.4 Note on Setting ACKBT in Master Receive Mode ... 770
17.7.5 Note on the States of Bits MST and TRN when Arbitration is Lost ... 771
17.7.6 Note on IICRST and BBSY bits ... 771

Section 18 Serial Sound Interface (SSI) ... 773
18.1 Features ... 773
18.2 Input/Output Pins ... 775
18.3 Register Description ... 776
18.3.1 Control Register (SSICR) ... 777
18.3.2 Status Register (SSISR) ... 783
18.3.3 Transmit Data Register (SSITDR) ... 788
18.3.4 Receive Data Register (SSIRDR) ... 788
18.4 Operation Description ... 789
18.4.1 Bus Format ... 789
18.4.2 Non-Compressed Modes ... 790
18.4.3 Operation Modes ... 800
18.4.4 Transmit Operation ... 801
18.4.5 Receive Operation ... 804
18.4.6 Temporary Stop and Restart Procedures in Transmit Mode ... 807
18.4.7 Serial Bit Clock Control ... 808
18.5 Usage Notes ... 809
18.5.1 Limitations from Overflow during Receive DMA Operation ... 809
18.5.2 Note on Using Oversample Clock ... 809
18.5.3 Restriction on Stopping Clock Supply ... 809

Section 19 Controller Area Network (RCAN-ET) ... 811
19.1 Summary ... 811
19.1.1 Overview ... 811
19.1.2 Scope ... 811
19.1.3 Audience ... 811
19.1.4 References ... 812

19.1.5 Features ... 812
19.2 Architecture ... 813
19.2.1 Block Diagram ... 813
19.2.2 Functions of Each Block ... 814
19.2.3 Input/Output Pins ... 815
19.2.4 Memory Map ... 816
19.3 Mailbox ... 817
19.3.1 Mailbox Structure ... 817
19.3.2 Message Control Field ... 819
19.3.3 Local Acceptance Filter Mask (LAFM) ... 823
19.3.4 Message Data Fields ... 824
19.4 RCAN-ET Control Registers ... 825
19.4.1 Master Control Register (MCR) ... 825
19.4.2 General Status Register (GSR) ... 831
19.4.3 Bit Configuration Register (BCR0, BCR1) ... 834
19.4.4 Interrupt Request Register (IRR) ... 839
19.4.5 Interrupt Mask Register (IMR) ... 845
19.4.6 Transmit Error Counter (TEC) and Receive Error Counter (REC) ... 846
19.5 RCAN-ET Mailbox Registers ... 847
19.5.1 Transmit Pending Register (TXPR0, TXPR1) ... 848
19.5.2 Transmit Cancel Register 0 (TXCR0) ... 851
19.5.3 Transmit Acknowledge Register 0 (TXACK0) ... 852
19.5.4 Abort Acknowledge Register 0 (ABACK0) ... 853
19.5.5 Data Frame Receive Pending Register 0 (RXPR0) ... 854
19.5.6 Remote Frame Receive Pending Register 0 (RFPR0) ... 855
19.5.7 Mailbox Interrupt Mask Register 0 (MBIMR0) ... 856
19.5.8 Unread Message Status Register 0 (UMSR0) ... 857
19.6 Application Note ... 858
19.6.1 Configuration of RCAN-ET ... 858
19.6.2 Test Mode Settings ... 863
19.6.3 Message Transmission Sequence ... 865
19.6.4 Message Receive Sequence ... 867
19.6.5 Reconfiguration of Mailbox ... 869
19.7 Interrupt Sources ... 871
19.8 CAN Bus Interface ... 873
19.9 Usage Notes ... 874
19.9.1 Module Standby Mode ... 874
19.9.2 Reset ... 874
19.9.3 CAN Sleep Mode ... 874
19.9.4 Register Access ... 874

19.9.5 Interrupts ... 875

Section 20 A/D Converter (ADC) ... 877
20.1 Features ... 877
20.2 Input/Output Pins ... 879
20.3 Register Configuration ... 880
20.3.1 A/D Data Registers A to H (ADDRA to ADDRH) ... 880
20.3.2 A/D Control/Status Register (ADCSR) ... 882
20.4 Operation ... 886
20.4.1 Single Mode ... 886
20.4.2 Multi Mode ... 889
20.4.3 Scan Mode ... 891
20.4.4 A/D Converter Activation by External Trigger, MTU2, or TMR ... 894
20.4.5 Input Sampling and A/D Conversion Time ... 894
20.4.6 External Trigger Input Timing ... 896
20.5 Interrupt Sources and DMAC Transfer Request ... 897
20.6 Definitions of A/D Conversion Accuracy ... 897
20.7 Usage Notes ... 899
20.7.1 Module Standby Mode Setting ... 899
20.7.2 Setting Analog Input Voltage ... 899
20.7.3 Notes on Board Design ... 899
20.7.4 Processing of Analog Input Pins ... 900
20.7.5 Permissible Signal Source Impedance ... 901
20.7.6 Influences on Absolute Precision ... 902
20.7.7 Note on Usage in Scan Mode and Multi Mode ... 902

Section 21 D/A Converter (DAC) ... 903
21.1 Features ... 903
21.2 Input/Output Pins ... 904
21.3 Register Descriptions ... 904
21.3.1 D/A Data Registers 0 and 1 (DADR0 and DADR1) ... 905
21.3.2 D/A Control Register (DACR) ... 905
21.4 Operation ... 907
21.5 Usage Notes ... 908
21.5.1 Module Standby Mode Setting ... 908
21.5.2 D/A Output Hold Function in Software Standby Mode ... 908
21.5.3 D/A Conversion and D/A Output in Deep Standby Mode ... 908
21.5.4 Setting Analog Input Voltage ... 908

Section 22 I/O Ports 909
22.1 Port A 909
22.1.1 Register Configuration 910
22.1.2 Port A Data Registers H and L (PADRH and PADRL) 910
22.1.3 Port A Port Registers H and L (PAPRH and PAPRL) 912
22.2 Port B 913
22.2.1 Register Configuration 914
22.2.2 Port B Data Registers H and L (PBDRH and PBDRL) 914
22.2.3 Port B Port Registers H and L (PBPRH and PBPRL) 916
22.3 Port C 917
22.3.1 Register Configuration 917
22.3.2 Port C Data Registers H and L (PCDRH and PCDRL) 918
22.3.3 Port C Port Registers H and L (PCPRH and PCPRL) 919
22.4 Port D 920
22.4.1 Register Configuration 920
22.4.2 Port D Data Register (PDDR) 921
22.4.3 Port D Port Registers H and L (PDPRH and PDPRL) 922
22.5 Port E 923
22.5.1 Register Configuration 923
22.5.2 Port E Port Register (PEPR) 923
22.6 Port F 924
22.6.1 Register Configuration 924
22.6.2 Port F Data Register (PFDR) 925
22.6.3 Port F Port Register (PFPR) 926

Section 23 Pin Function Controller (PFC) 927
23.1 Register Descriptions 935
23.1.1 Port A I/O Registers H and L (PAIORH and PAIORL) 937
23.1.2 Port A Control Registers 1 to 8 (PACR1 to PACR8) 938
23.1.3 Port B I/O Registers H and L (PBIORH and PBIORL) 948
23.1.4 Port B Control Registers 1 to 8 (PBCR1 to PBCR8) 949
23.1.5 Port C I/O Registers H and L (PCIORH and PCIORL) 962
23.1.6 Port C Control Registers 1 to 7 (PCCR1 to PCCR7) 963
23.1.7 Port D I/O Register (PDIOR) 973
23.1.8 Port D Control Registers 1 to 5 (PDCR1 to PDCR5) 974
23.1.9 Port E Control Registers 1 and 2 (PECR1 and PECR2) 981
23.1.10 Port F I/O Register (PFIOR) 983
23.1.11 Port F Control Registers 1 and 2 (PFCR1 and PFCR2) 984
23.2 Usage Note 988

Section 24 On-Chip RAM 989
24.1 Features 989
24.2 Usage Notes 990
24.2.1 Page Conflict 990
24.2.2 RAME and RAMWE Bits 990

Section 25 Power-Down Modes 991
25.1 Features 991
25.1.1 Power-Down Modes 991
25.2 Register Descriptions 993
25.2.1 Standby Control Register (STBCR) 994
25.2.2 Standby Control Register 2 (STBCR2) 995
25.2.3 Standby Control Register 3 (STBCR3) 997
25.2.4 Standby Control Register 4 (STBCR4) 998
25.2.5 Standby Control Register 5 (STBCR5) 1000
25.2.6 System Control Register 1 (SYSCR1) 1002
25.2.7 System Control Register 2 (SYSCR2) 1003
25.2.8 RAM Retaining Area Specifying Register (RAMKP) 1004
25.2.9 Deep Standby Oscillation Settling Clock Select Register (DSCNT) 1005
25.2.10 Deep Standby Cancel Source Flag Register (DSFR) 1006
25.3 Operation 1008
25.3.1 Sleep Mode 1008
25.3.2 Software Standby Mode 1009
25.3.3 Software Standby Mode Application Example 1011
25.3.4 Deep Standby Mode 1012
25.3.5 Module Standby Function 1017
25.4 Usage Note 1017
25.4.1 Note on Setting Registers 1017
25.4.2 Note on Canceling Standby Mode when an External Clock is being Input 1017

Section 26 User Debugging Interface (H-UDI) 1019
26.1 Features 1019
26.2 Input/Output Pins 1020
26.3 Register Descriptions 1021
26.3.1 Bypass Register (SDBPR) 1021
26.3.2 Instruction Register (SDIR) 1022
26.4 Operation 1023
26.4.1 TAP Controller 1023
26.4.2 Reset Types 1024
26.4.3 UDTDO Output Timing 1024

26.4.4 H-UDI Reset 1025
26.4.5 H-UDI Interrupt 1025
26.5 Usage Notes 1026

Section 27 Advanced User Debugger II (AUD-II) 1027
27.1 Features 1027
27.2 Input/Output Pins 1027
27.3 RAM Monitor Mode 1029
27.3.1 Communication Protocol 1029
27.3.2 Operation 1030
27.3.3 Usage Notes (RAM Monitor Mode) 1032

Section 28 List of Registers 1033
28.1 Register Addresses (Address Order) 1034
28.2 Register Bits 1052
28.3 Register States in Each Operating Mode 1090

Section 29 Electrical Characteristics 1107
29.1 Absolute Maximum Ratings 1107
29.2 DC Characteristics 1108
29.3 AC Characteristics 1116
29.3.1 Clock Timing 1116
29.3.2 Control Signal Timing 1120
29.3.3 Bus Timing 1122
29.3.4 DMAC Module Timing 1136
29.3.5 UBC Trigger Timing 1137
29.3.6 MTU2 Module Timing 1138
29.3.7 8-Bit Timer Timing 1139
29.3.8 Watchdog Timer Timing 1140
29.3.9 SCIF Module Timing 1141
29.3.10 IIC3 Module Timing 1142
29.3.11 SSI Module Timing 1143
29.3.12 RCAN-ET Module Timing 1145
29.3.13 A/D Trigger Input Timing 1146
29.3.14 I/O Port Timing 1146
29.3.15 H-UDI-Related Pin Timing 1147
29.3.16 AUD-II Timing 1149
29.3.17 AC Characteristics Measurement Conditions 1150
29.4 A/D Converter Characteristics 1151
29.5 D/A Converter Characteristics 1152

29.6 Usage Note 1153

Appendix 1155
A. Pin States 1155
B. Package Dimensions 1160

Main Revisions for This Edition 1161

Index 1183

Section 1 Overview

1.1 SH7201 Group Features

This LSI is a single-chip RISC (Reduced Instruction Set Computer) microprocessor that integrates a Renesas original RISC CPU core with peripheral functions required for system configuration.

The CPU incorporated in this LSI is the SH-2A CPU, which features upward compatibility on the object code level with the SH-1, SH-2, and SH-2E microcomputers. The CPU has a RISC-type instruction set and employs a superscalar architecture and the Harvard architecture, which greatly improves instruction execution speed. In addition, the 32-bit internal-bus architecture independent of the bus for the direct memory access controller (DMAC) enhances data processing power. This CPU realizes low-cost, high-performance, and high-functioning systems for applications such as high-speed realtime control, which has been next to impossible with the conventional microcomputers.

This LSI has a floating-point unit and a cache.

In addition, this LSI includes on-chip peripheral functions necessary for system configuration, such as, 32-Kbyte RAM for high-speed operation, a controller area network (RCAN-ET), a serial sound interface (SSI), a serial communication interface with FIFO (SCIF), I^2C bus interface 3 (IIC3), a multi-function timer pulse unit 2 (MTU2), an 8-bit timer (TMR), a realtime clock (RTC), an A/D converter, a D/A converter, an interrupt controller (INTC), I/O ports, and advanced user debugger II (AUD-II).

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 SH7201 Group Features

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

Item	Features
Floating-point Unit (FPU)	• Floating-point co-processor included • Supports single-precision (32-bit) and double-precision (64-bit) • Supports data type and exceptions that conforms to IEEE754 standard • Two rounding modes: Round to nearest and round to zero • Denormalization modes: Flush to zero • Floating-point registers Sixteen 32-bit floating-point registers (single-precision × 16 words or double-precision × 8 words) Two 32-bit floating-point system registers • Supports FMAC (multiplication and accumulation) instructions • Supports FDIV (division) and FSQRT (square root) instructions • Supports FLDI0/FLDI1 (load constant 0/1) instructions • Instruction execution time Latency (FAMC/FADD/FSUB/FMUL): Three cycles (single-precision), eight cycles (double-precision) Pitch (FAMC/FADD/FSUB/FMUL): One cycle (single-precision), six cycles (double-precision) Note: FMAC only supports single-precision • 5-stage pipeline
Cache	• Instruction cache: 8 Kbytes • Operand cache: 8 Kbytes • 128-entry, 4-way set associative, 16-byte block length configuration each for the instruction cache and operand cache • Write-back, write-through and LRU replacement algorithm • Cache locking function available (only for operand cache); ways 2 and 3 can be locked
Interrupt controller (INTC)	• Seventeen external interrupt pins (NMI, IRQ7 to IRQ0, and PINT7 to PINT0) • On-chip peripheral interrupts: Priority level set for each module • 16 priority levels available • Register bank enabling fast register saving and restoring in interrupt handling

Item	Features
Bus state controller (BSC)	• CSC — Seven-channel chip select controller (CSC) — External devices with their bus sizes of 32, 16, or 8 bits can be connected — Cycle wait function Up to 31 cycles (up to 7 cycles for page access cycle) — The following features settable for wait controlling Timings of asserting and negating chip select signals Timings of asserting and negating read/write signals Timings of starting and stopping data output — One-write strobe and byte write strobe modes are available as write access modes — Page read and page write modes are available as page access modes
	• SDRAMC — Two-channel external SDRAM interfaces — Auto refresh using the internal programmable refresh counter or self refresh mode selectable — The following features settable Row-column latency, column latency, row-active period, write-recovery period, row precharge period, auto refresh request interval, initial precharge cycle count, and initial auto refresh request interval — Random column burst access available (one SDRAM burst length) — Initialization sequencer issues precharge and auto refresh commands
Bus monitor	• Bus monitor function When an illegal address access or a bus timeout is detected, a bus error interrupt is generated.

Item	Features
Direct memory access controller (DMAC)	• Eight channels; external request available for four of them • Can be activated by software, on-chip modules, or external devices — Software; 1, internal source; 32, external source; 4 • Up to 64 Mbytes can be transferred • Maximum transfer data size — 8, 16, or 32 bits for single-data transfer — 1, 2, 4, 8, 16, 32, 64, or 128 sets of data for single operand transfer (a transfer continues until the byte count reaches 0) • Transfer method — Cycle-stealing transfer (dual address transfer) Three clock cycles per one set of data (best) Bus released between read and write cycles — Pipeline transfer (dual address transfer) One clock cycle per one set of data (best) • Addressing method Increment, decrement, or fixed • Three clock cycles per one set of data (best) • Transfer modes Single operand transfer, continuous operand transfer, and non-stop transfer • An interrupt is requested when the byte count reaches 0 • Reloading function Source address, destination address, and byte count • DMAC suspend, resume, and stop function • DMAC forcible terminate function
Clock pulse generator (CPG)	• Clock mode: Input clock can be selected from external input (EXTAL or CKIO) or crystal resonator • Input clock can be multiplied by 16 (max.) by the internal PLL circuit • Three types of clocks generated CPU clock: Maximum 120 MHz Bus clock: Maximum 60 MHz Peripheral clock: Maximum 40 MHz
Watchdog timer (WDT)	• On-chip one-channel watchdog timer • A counter overflow can reset this LSI

Item	Features
Power-down modes	• Four power-down modes provided to reduce the current consumption in this LSI — Sleep mode — Software standby mode — Deep standby mode — Module standby mode
Multi-function timer pulse unit 2 (MTU2)	• Maximum 16 lines of pulse inputs/outputs and 3 lines of pulse inputs based on six channels of 16-bit timers • 21 output compare and input capture registers • Input capture function • Pulse output modes One shot, 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 cycle specifiable — A/D converter start request 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
8-bit timer (TMR)	• Two-channel 8-bit timer • Six internal clocks (Pϕ/2, Pϕ/8, Pϕ/32, Pϕ/64, Pϕ/1024, or Pϕ/8192) or external clock specifiable • Timer outputs controllable using two compare match signals • Two channels can be cascade-connected
Realtime clock (RTC)	• Internal clock, calendar function, alarm function • Interrupts can be generated at intervals of 1/256 s by the 32.768-kHz on-chip crystal oscillator

Item	Features
Serial communication interface with FIFO (SCIF)	• Eight channels • Clock 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
I^2C bus interface 3 (IIC3)	• Three channels • Master mode and slave mode supported
Serial sound interface (SSI)	• Two-channel bidirectional serial transfer • Support of various serial audio formats • Support of master and slave functions • Generation of programmable word clock and bit clock • Multichannel formats • Support of 8, 16, 18, 20, 22, 24 and 32-bit data formats
Controller area network (RCAN-ET)	• Two channels • Supports CAN specification 2.0B — Data and remote frame in standard format (11-bit ID) — Data and remote frame in extended format (18-bit ID) • 16 independent message buffers using IDs in standard (11-bit) or extended (18-bit) format • 15 Mailboxes for transmission or reception • One receive-only Mailbox • Message reception filtering by IDs: — Standard message ID — Extended message ID • Local reception filter for all Mailboxes (standard and extended IDs) can be specified • Power consumption can be reduced in sleep mode • CAN data transfer rate of up to 1 Mbit/s available • Transmit message queue having an internal priority sorting mechanism which handles priority-inversion issue of realtime applications • Data buffer access without hand-shaking
I/O ports	• 109 I/Os and 14 inputs • Input or output can be selected for each bit

Item	Features
A/D converter (ADC)	• 10-bit resolution • Eight input 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)	• Two break channels • Addresses, data values, type of access, and data size can all be set as break conditions
User debugging interface (H-UDI)	• E10A emulator support • JTAG-standard pin assignment
Advanced user debugger II	• Eight I/O pins • Functions to read/write modules connected to internal/external buses (except cache and H-UDI) in RAM monitor mode
On-chip RAM	• 32-Kbyte memory
Power supply voltage	• PVcc, VccR, and PLLVcc: 3.0 to 3.6 V
Packages	• LQFP2424-176Cu (0.5 pitch)

1.2 Product Lineup

Table 1.2 Product Lineup

Abbreviation	Product Code	Operating Temperature
R5S72011	R5S72011RB120FP	−20 to +70°C (Regular specifications)
	R5S72011RW100FP	−20 to +85°C (Wide-range specifications)

1.3 Block Diagram

The block diagram of this LSI is shown in figure 1.1.

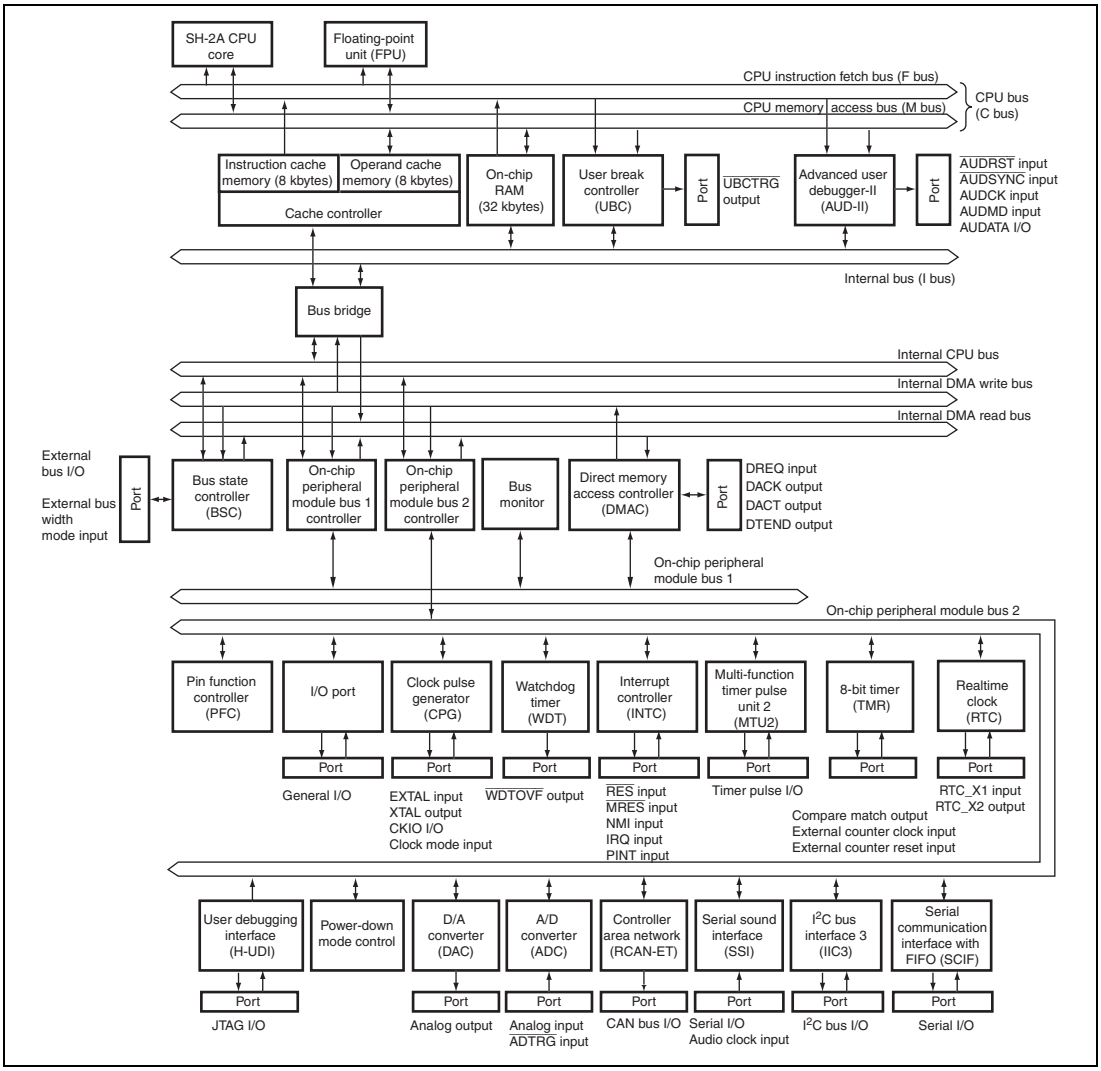


Figure 1.1 Block Diagram

1.4 Pin Assignments

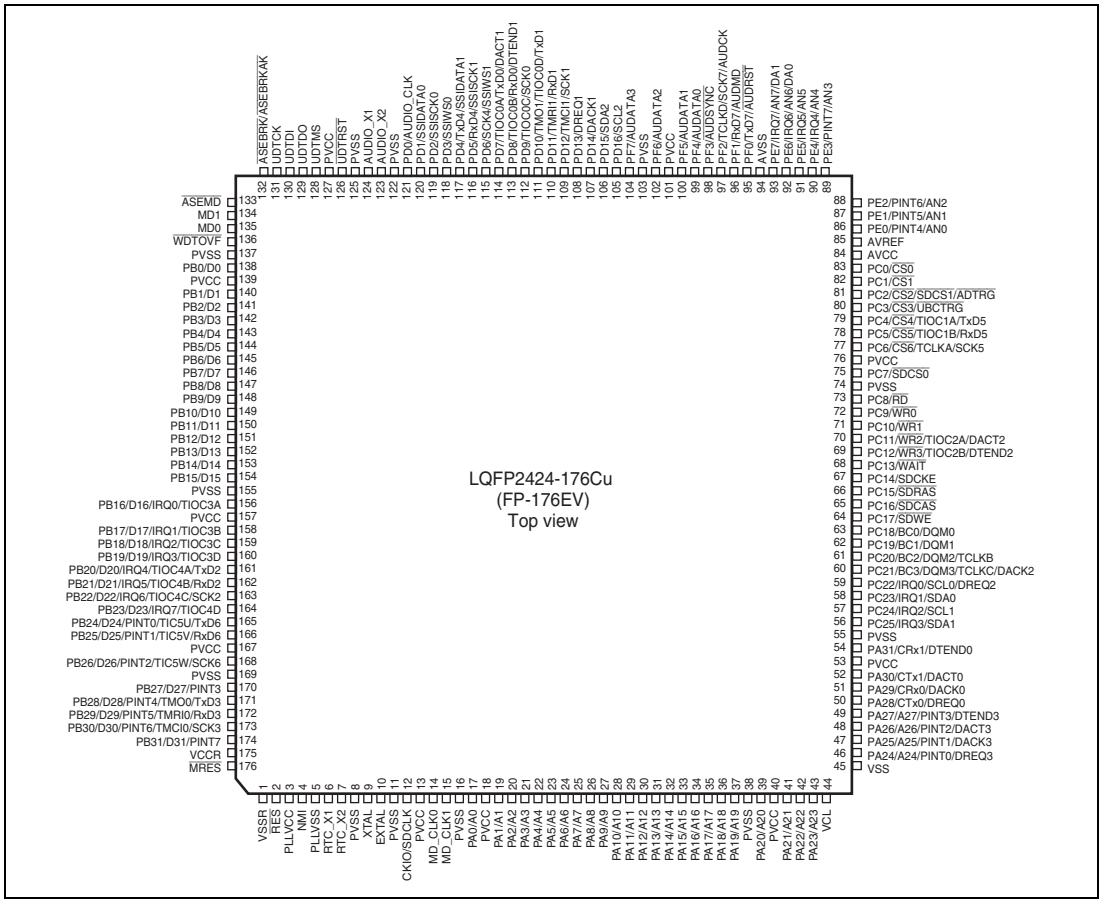


Figure 1.2 Pin Assignments

1.5 Pin Functions

Table 1.3 lists the pin functions.

Table 1.3 Pin Functions

Classification	Symbol	I/O	Name	Function
Power supply	VCCR	I	Power supply for internal step-down circuit	Power supply pin for the internal step-down circuit. This pin must be connected to the system power supply. This LSI does not operate correctly if this pin is left open.
	VSSR	I	Ground for internal step-down circuit	Ground pin for the internal step-down circuit. This pin must be connected to the system power supply (0 V). This LSI does not operate correctly if this pin is left open.
	VCL	I	Capacitor connected pin for internal step-down circuit	Pin for connecting an external capacitor for the internal step-down circuit. This pin should be connected to the VSS via the external capacitor (place closer to this pin).
	VSS	I	Ground for internal step-down circuit	Ground pin for the internal step-down circuit used for stabilize internal step-down power supply. This pin should be connected to the VCL via the external capacitor (place closer to this pin)
	PVCC	I	Power supply for I/O circuits	Power supply pins for I/O pins. All the PVCC pins must be connected to the system power supply. This LSI does not operate correctly if there is a pin left open.
	PVSS	I	Ground for I/O circuits	Ground pins for I/O pins. All the PVSS pins must be connected to the system power supply (0 V). This LSI does not operate correctly if there is a pin left open.

Classification	Symbol	I/O	Name	Function
Power supply	PLLVCC	I	Power supply for PLL	Power supply for the on-chip PLL oscillator. This LSI does not operate correctly if this pin is left open.
	PLLVSS	I	Ground for PLL	Ground pin for the on-chip PLL oscillator. This LSI does not operate correctly if this pin is left open.
Clock	EXTAL	I	Crystal resonator/ external clock	Pin connected to a crystal resonator. An external clock signal may also be input to the EXTAL pin.
	XTAL	O		
	CKIO	I/O	System clock I/O	Input pin for an external clock or output pin for supplying the system clock to external devices
Operating mode control	MD1, MD0	I	Mode set	Pins to set the operating mode. Do not change signal levels on these pins during operation.
	MD_CLK1, MD_CLK0	I	Clock mode set	Pins to set the clock operating mode. Do not change signal levels on these pins during operation.
	$\overline{\text{ASEMD}}$	I	Debugging mode	This pin is valid when the E10A-USB emulator is in use. Otherwise, fix the signal level on this pin high.
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	An overflow signal from the WDT is output on this pin.
	$\overline{\text{ASEBRKAK}}$	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

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.
	PINT7 to PINT0	I	Interrupt requests 7 to 0	Maskable interrupt request pins. Only level-input detection can be selected.
Address bus	A27 to A0	O	Address bus	Addresses are output on these pins.
Data bus	D31 to D0	I/O	Data bus	Bidirectional data bus
Bus control	$\overline{CS6}$ to $\overline{CS0}$	O	Chip select 6 to 0	Chip-select signals for external memory or devices
	$\overline{RD}$	O	Read	Indicates that data is read from an external device.
	$\overline{WAIT}$	I	Wait	Input pin for inserting a wait cycle into the bus cycles during access to the external space
	$\overline{WR0}$	O	Byte select	Indicates a write access to bits 7 to 0 of data of external memory or device. (For an access in units of 8, 16, or 32 bits)
	$\overline{WR1}$	O	Byte select	Indicates a write access to bits 15 to 8 of data of external memory or device. (For an access in units of 16 or 32 bits)
	$\overline{WR2}$	O	Byte select	Indicates a write access to bits 23 to 16 of data of external memory or device. (For an access in units of 32 bits)
	$\overline{WR3}$	O	Byte select	Indicates a write access to bits 31 to 24 of data of external memory or device. (For an access in units of 32 bits)

Classification	Symbol	I/O	Name	Function
Bus control	$\overline{BC0}$	O	Byte select	Selects bits 7 to 0 of data of external memory or device. (For an access in units of 8, 16, or 32 bits)
	$\overline{BC1}$	O	Byte select	Selects bits 15 to 8 of data of external memory or device. (For an access in units of 16 or 32 bits)
	$\overline{BC2}$	O	Byte select	Selects bits 23 to 16 of data of external memory or device. (For an access in units of 32 bits)
	$\overline{BC3}$	O	Byte select	Selects bits 31 to 24 of data of external memory or device. (For an access in units of 32 bits)
	DQM0	O	Byte select	Selects bits D7 to D0 when SDRAM is connected. (For an access in units of 8, 16, or 32 bits)
	DQM1	O	Byte select	Selects bits D15 to D8 when SDRAM is connected. (For an access in units of 16 or 32 bits)
	DQM2	O	Byte select	Selects bits D23 to D16 when SDRAM is connected. (For an access in units of 32 bits)
	DQM3	O	Byte select	Selects bits D31 to D24 when SDRAM is connected. (For an access in units of 32 bits)
	$\overline{SDCS1}$, $\overline{SDCS0}$	O	Chip select	Pins connected to the $\overline{CS}$ pins of SDRAM
	$\overline{SDRAS}$	O	RAS	Pin connected to the $\overline{RAS}$ pin of SDRAM
	$\overline{SDCAS}$	O	CAS	Pin connected to the $\overline{CAS}$ pin of SDRAM
	$\overline{SDWE}$	O	WE	Pin connected to the $\overline{WE}$ pin of SDRAM
	SDCKE	O	CK enable	Pin connected to the CKE pin of SDRAM
	SDCLK	O	Clock output	Pin connected to the CLK pin of SDRAM

Classification	Symbol	I/O	Name	Function
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 acknowledge	Output pins for signals indicating acknowledge of external requests from external devices
	DACT3 to DACT0	O	DMA-transfer request active	Output pins for signals indicating DMA active in response to external requests from external devices
	DTEND3 to DTEND0	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.
	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.
	TIOC5U, TIOC5V, TIOC5W	I	MTU2 input capture (channel 5)	The TGRU_5, TGRV_5, and TGRW_5 input capture input/dead time compensation input pins.

Classification	Symbol	I/O	Name	Function
8-bit timer (TMR)	TMO0, TMO1	O	Timer output	Pins for waveform outputs by output compare
	TMCI0, TMCI1, TMRI0, TMRI1	I	Timer clock/timer reset input	Input pins for an external clock or an external reset for the timer
Realtime clock (RTC)	RTC_X1	I	Crystal resonator for RTC	Pin connected to 32.768-kHz crystal resonator
	RTC_X2	O		
Serial communication interface with FIFO (SCIF)	TxD7 to TxD0	O	Transmit data	Data output pins
	RxD7 to RxD0	I	Receive data	Data input pins
	SCK7 to SCK0	I/O	Serial clock	Clock input/output pins
I^2C bus interface 3 (IIC3)	SCL2 to SCL0	I/O	Serial clock pin	Serial clock input/output pin
	SDA2 to SDA0	I/O	Serial data pin	Serial data input/output pin
Serial sound interface (SSI)	SSIDATA0, SSIDATA1	I/O	SSI data I/O	I/O pins for serial data
	SSISCK0, SSISCK1	I/O	SSI clock I/O	I/O pins for serial clocks
	SSIWS0, SSIWS1	I/O	SSI clock LR I/O	I/O pins for word selection
	AUDIO_CLK	I	External clock for SSI audio	Input pin of external clock for SSI audio (32/44.1/48 kHz × 256/384/512). A clock input to the divider is selected from an oscillation clock input on this pin or pins AUDIO_X1 and AUDIO_X2.
	AUDIO_X1	I	Crystal resonator for SSI audio	Pins connected to a crystal resonator for SSI audio. An external clock can be input on pin AUDIO_X1 (32/44.1/48 kHz × 256/384/512). A clock input to the divider is selected from an oscillation clock input on these pins or the AUDIO_CLK pin.
	AUDIO_X2	O		

Classification	Symbol	I/O	Name	Function
Controller area network (RCAN-ET)	CTx0, CTx1	O	CAN bus transmit data	Output pin for transmit data on the CAN bus
	CRx0, CRx1	I	CAN bus receive data	Output pin for receive data on the CAN bus
A/D converter	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
D/A converter	DA1, DA0	O	Analog output pins	Analog output pins
Analog power supply	AVcc	I	Analog power supply	Power supply pins for the A/D converter and D/A converter
	AVref	I	Analog reference power supply	Reference voltage input pin for the A/D converter and D/A converter
	AVss	I	Analog ground	Ground pins for the A/D converter and D/A converter
I/O ports	PA31 to PA0	I/O	General port	32-bit general I/O port pins
	PB31 to PB0	I/O	General port	32-bit general I/O port pins
	PC25 to PC22	I	General port	4-bit general input port pins
	PC21 to PC0	I/O	General port	22-bit general I/O port pins
	PD16 to PD15	I	General port	2-bit general input port pins
	PD14 to PD0	I/O	General port	15-bit general I/O port pins
	PE7 to PE0	I	General port	8-bit general input port pins
	PF7 to PF0	I/O	General port	8-bit general I/O port pins
User debugging interface (H-UDI)	UDTCK*	I	Test clock	Test-clock input pin
	UDTMS*	I	Test mode select	Test-mode select signal input pin
	UDTDI*	I	Test data input	Serial input pin for instructions and data
	UDTDO	O	Test data output	Serial output pin for instructions and data
	$\overline{\text{UDTRST}}$*	I	Test reset	Initialization-signal input pin

Classification	Symbol	I/O	Name	Function
Advanced user debugger II (AUD-II)	AUDATA3 to AUDATA0	I/O	AUD data	Input pins for monitor addresses/data I/O pins
	AUDCK	I	AUD clock	External clock input pin
	$\overline{\text{AUDSYNC}}$	I	AUD sync signal	Input pin for an signal identifying the data start position
	AUDMD	I	AUD mode	Pin to select the AUD mode
	$\overline{\text{AUDRST}}$	I	AUD reset	Input pins for an AUD reset
User break controller (UBC)	$\overline{\text{UBCTRG}}$	O	User break trigger output	Trigger output pin for UBC condition match

Note: * The pin with the pull-up function.

Section 2 CPU

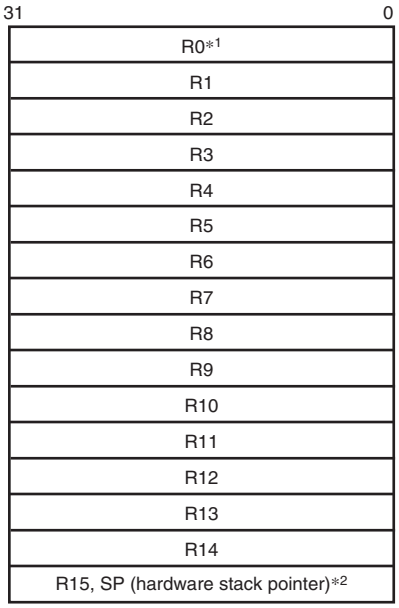
2.1 Register Configuration

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.

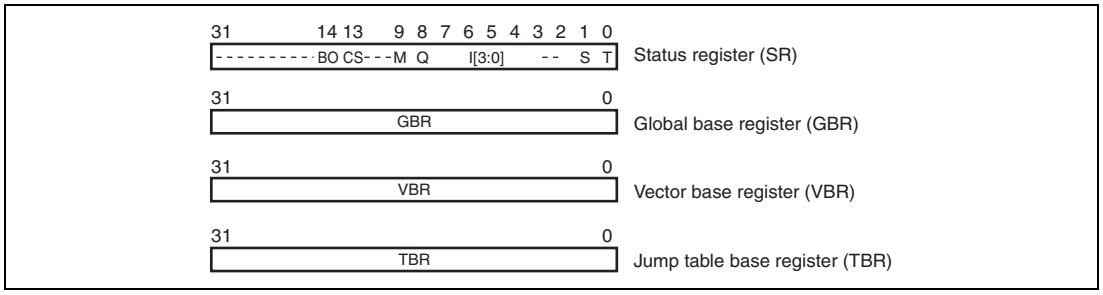


Figure 2.2 Control Registers

(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 points four bytes ahead of the current instruction and controls the flow of the processing.

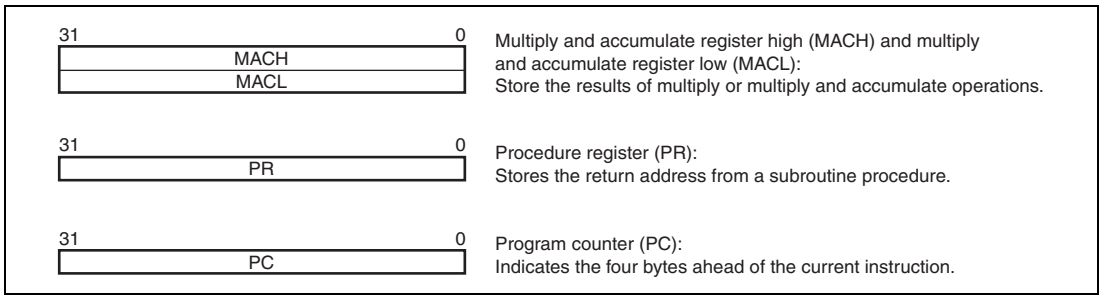


Figure 2.3 System Registers

(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 points four bytes ahead 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.

This LSI has 15 banks. For details, see the SH-2A, SH2A-FPU Software Manual and section 6.8, Register Banks.

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.

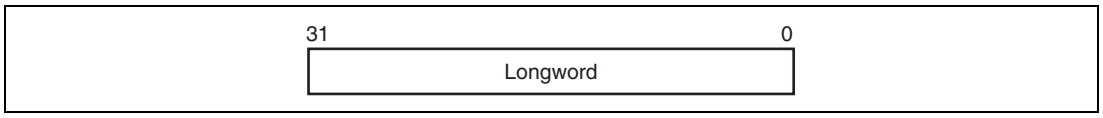


Figure 2.4 Data Format in Registers

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.

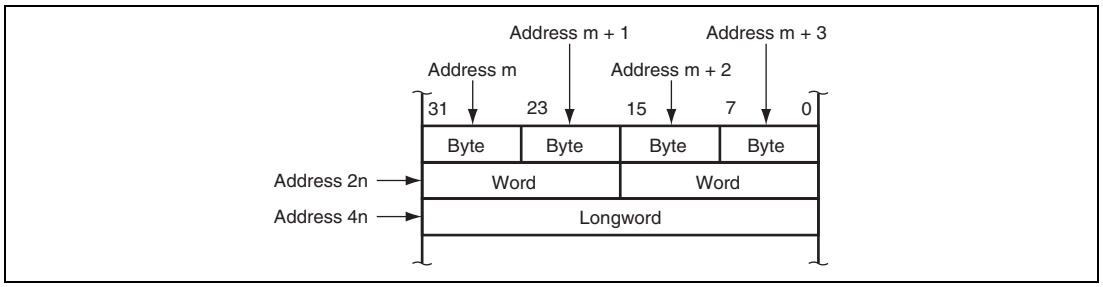


Figure 2.5 Data Formats in Memory

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

SH-2A 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.

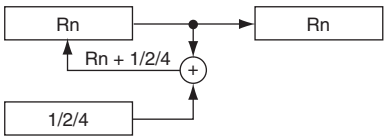
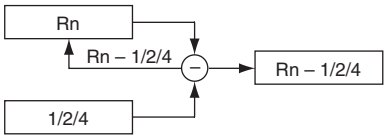
Table 2.7 Displacement Accessing

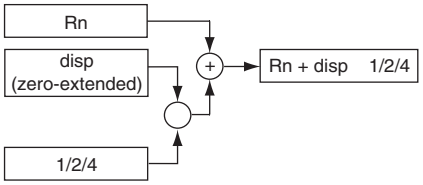
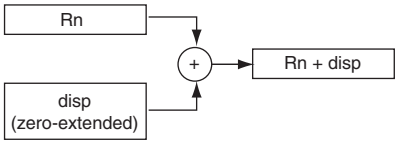
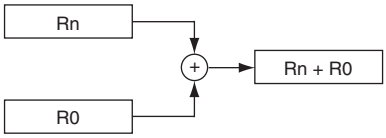
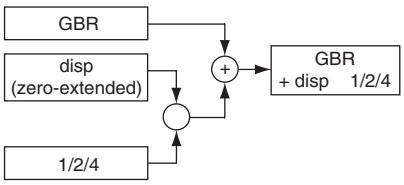
Classification	SH-2A CPU		Example of Other CPU	
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W	@(H'1234,R1),R2
	MOV.W	@(R0,R1),R2		
		.....		
	.DATA.W	H'1234		

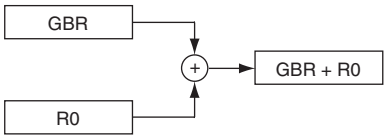
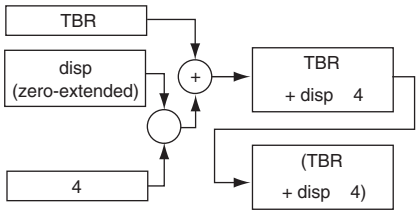
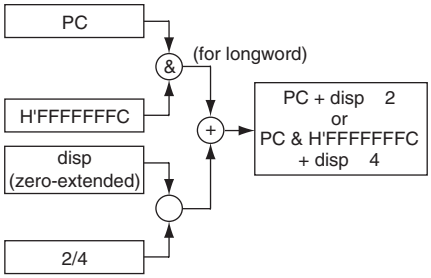
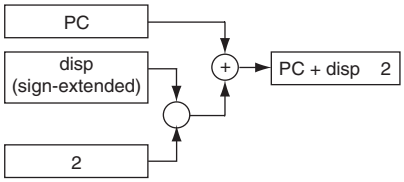
2.3.2 Addressing Modes

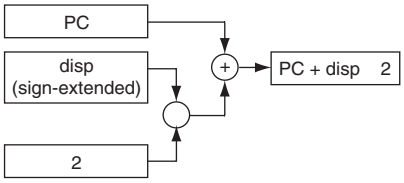
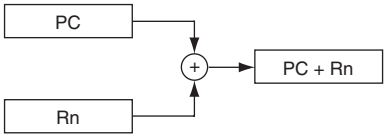
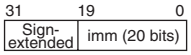
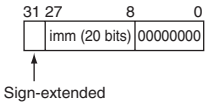
Addressing modes and effective address calculation are as follows:

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
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

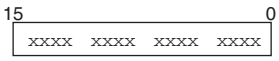
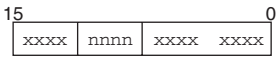
Addressing Mode	Instruction Format	Effective Address Calculation	Equation
PC relative	disp:12	The effective address is the sum of PC value and the value that is obtained by doubling the sign-extended 12-bit displacement (disp). 	PC + disp × 2
	Rn	The effective address is the sum of PC value and Rn. 	PC + Rn
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.	—

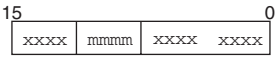
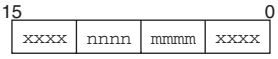
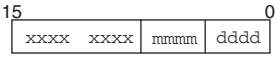
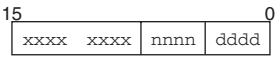
2.3.3 Instruction Format

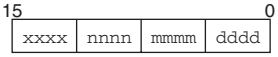
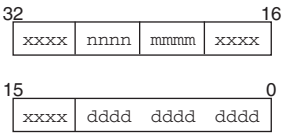
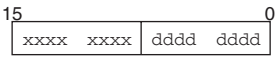
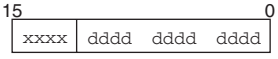
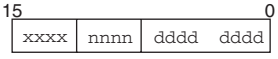
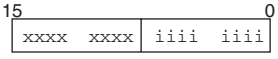
The instruction formats and the meaning of source and destination operands are described below. The meaning of the operand depends on the instruction code. The symbols used are as follows:

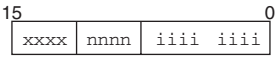
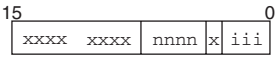
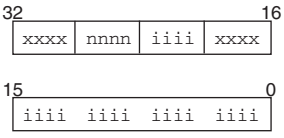
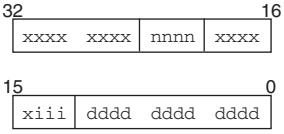
- xxxx: Instruction code
- mmmm: Source register
- nnnn: Destination register
- iiii: Immediate data
- dddd: Displacement

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`
nd4 format 	R0 (Register direct)	nnnndddd: Register indirect with displacement	`MOV.B R0,@(disp,Rn)`

Instruction Formats	Source Operand	Destination Operand	Example
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`
i format 	iiiiiiii: Immediate	Indexed GBR indirect	`AND.B #imm,@(R0,GBR)`
	iiiiiiii: Immediate	R0 (Register direct)	`AND #imm,R0`
	iiiiiiii: Immediate	—	`TRAPA #imm`

Instruction Formats	Source Operand	Destination Operand	Example
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 	nnnndddddddddddd: Register indirect with displacement iii: Immediate	—	BLD.B #imm3,@(disp12,Rn)
	—	nnnndddddddddddd: Register indirect with displacement iii: Immediate	BST.B #imm3,@(disp12,Rn)

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

2.4 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 Immediate data transfer Peripheral module data transfer Structure data transfer Reverse stack transfer	62
		MOVA	Effective address transfer	
		MOVI20	20-bit immediate data transfer	
		MOVI20S	20-bit immediate data transfer 8-bit left-shit	
		MOVML	R0–Rn register save/restore	
		MOVMU	Rn–R14 and PR register save/restore	
		MOVRT	T bit inversion and transfer to Rn	
		MOVT	T bit transfer	
		MOVU	Unsigned data transfer	
		NOTT	T bit inversion	
		PREF	Prefetch to operand cache	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	

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

Classification	Types	Operation Code	Function	No. of Instructions
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	12	ROTL	One-bit left rotation	16
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAD	Dynamic arithmetic shift	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLD	Dynamic logical shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
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	
Floating-point instructions	19	FABS	Floating-point absolute value	48
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		FCNVDS	Conversion from double-precision to single-precision	
		FCNVSD	Conversion from single-precision to double-precision	
		FDIV	Floating-point division	
		FLDI0	Floating-point load immediate 0	
		FLDI1	Floating-point load immediate 1	
		FLDS	Floating-point load into system register FPUL	
		FLOAT	Conversion from integer to floating-point	
		FMAC	Floating-point multiply and accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	

Classification	Types	Operation Code	Function	No. of Instructions
Floating-point instructions	19	FSCHG	SZ bit inversion	48
		FSQRT	Floating-point square root	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Bit manipulation	10	BAND	Bit AND	14
		BCLR	Bit clear	
		BLD	Bit load	
		BOR	Bit OR	
		BSET	Bit set	
		BST	Bit store	
		BXOR	Bit exclusive OR	
		BANDNOT	Bit NOT AND	
		BORNOT	Bit NOT OR	
		BLDNOT	Bit NOT load	
Total:	112			253

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

Instruction	Instruction Code	Operation	Execution 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.
Explanation of Symbols Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement*2	Explanation of Symbols mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 ..... 1111: R15 iiii: Immediate data dddd: Displacement	Explanation of Symbols →, ←: 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
MOV.W	Rm,@(R0,Rn)	0000nnnnmmmm0101	Rm → (R0 + Rn)	1	—	Yes	Yes	Yes

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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
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 Rn 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 Rn as PR	1 to 16	—			Yes
MOVRT	Rn	0000nnnn00111001	~T → Rn	1	—			Yes
MOVT	Rn	0000nnnn00101001	T → Rn	1	—	Yes	Yes	Yes
MOVU.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 1000dddddddddddd	(disp + Rm) → zero extension → Rn	1	—			Yes
MOVU.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 1001dddddddddddd	(disp × 2 + Rm) → zero extension → Rn	1	—			Yes

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility		
						SH2, SH2E	SH4	SH-2A
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	Overflow	Yes	Yes	Yes
CMP/EQ	#imm,R0	10001000iiiiiiii	When R0 = imm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	When Rn = Rm, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/HS	Rm,Rn	0011nnnnmmmm0010	When Rn ≥ Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/GE	Rm,Rn	0011nnnnmmmm0011	When Rn ≥ Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/HI	Rm,Rn	0011nnnnmmmm0110	When Rn > Rm (unsigned), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/GT	Rm,Rn	0011nnnnmmmm0111	When Rn > Rm (signed), 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/PL	Rn	0100nnnn00010101	When Rn > 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/PZ	Rn	0100nnnn00010001	When Rn ≥ 0, 1 → T Otherwise, 0 → T	1	Comparison result	Yes	Yes	Yes
CMP/STR	Rm,Rn	0010nnnnmmmm1100	When any bytes are equal, 1 → T Otherwise, 0 → T	1	Comparison 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
EXTU.B	Rm,Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended → Rn	1	—	Yes	Yes	Yes

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

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2, SH2E	SH4	SH-2A
STC	SR,Rn	0000nnnn00000010	SR → Rn	2	—	Yes	Yes	Yes
STC	TBR,Rn	0000nnnn01001010	TBR → Rn	1	—			Yes
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 Floating Point Operation Instructions

Table 2.17 Floating Point Operation Instructions

Instruction		Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn|→FRn	1	—	Yes	Yes	Yes
FABS	DRn	1111nnn001011101	|DRn|→DRn	1	—		Yes	Yes
FADD	FRm, FRn	1111nnnnmmmm0000	FRn+FRm→FRn	1	—	Yes	Yes	Yes
FADD	DRm, DRn	1111nnn0mmm00000	DRn+DRm→DRn	6	—		Yes	Yes
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn=FRm)? 1:0→T	1	Operation result	Yes	Yes	Yes
FCMP/EQ	DRm, DRn	1111nnn0mmm00100	(DRn=DRm)? 1:0→T	2	Operation result		Yes	Yes
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn>FRm)? 1:0→T	1	Operation result	Yes	Yes	Yes
FCMP/GT	DRm, DRn	1111nnn0mmm00101	(DRn>DRm)? 1:0→T	2	Operation result		Yes	Yes
FCNVDS	DRm, FPUL	1111mmm010111101	(float)DRm→FPUL	2	—		Yes	Yes
FCNVSD	FPUL, DRn	1111nnn010101101	(double)FPUL→DRn	2	—		Yes	Yes
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm→FRn	10	—	Yes	Yes	Yes
FDIV	DRm, DRn	1111nnn0mmm00011	DRn/DRm→DRn	23	—		Yes	Yes
FLDI0	FRn	1111nnnn10001101	0×00000000→FRn	1	—	Yes	Yes	Yes
FLDI1	FRn	1111nnnn10011101	0×3F800000→FRn	1	—	Yes	Yes	Yes
FLDS	FRm, FPUL	1111mmmm00011101	FRm→FPUL	1	—	Yes	Yes	Yes
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL→FRn	1	—	Yes	Yes	Yes
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL→DRn	2	—		Yes	Yes
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0×FRm+FRn→FRn	1	—	Yes	Yes	Yes
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm→FRn	1	—	Yes	Yes	Yes
FMOV	DRm, DRn	1111nnn0mmm01100	DRm→DRn	2	—		Yes	Yes
FMOV.S	@(R0, Rm), FRn	1111nnnnmmmm0110	(R0+Rm) →FRn	1	—	Yes	Yes	Yes
FMOV.D	@(R0, Rm), DRn	1111nnn0mmmm0110	(R0+Rm) →DRn	2	—		Yes	Yes
FMOV.S	@Rm+, FRn	1111nnnnmmmm1001	(Rm) →FRn, Rm+=4	1	—	Yes	Yes	Yes
FMOV.D	@Rm+, DRn	1111nnn0mmmm1001	(Rm) →DRn, Rm+=8	2	—		Yes	Yes

Instruction	Instruction Code	Operation	Execution Cycles	T Bit	Compatibility SH2E	SH4	SH-2A/ SH2A-FPU
FMOV.S @Rm, FRn	1111nnnnmmmm1000	(Rm) →FRn	1	—	Yes	Yes	Yes
FMOV.D @Rm, DRn	1111nnn0mmmm1000	(Rm) →DRn	2	—		Yes	Yes
FMOV.S @(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp×4+Rm) →FRn	1	—			Yes
FMOV.D @(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp×8+Rm) →DRn	2	—			Yes
FMOV.S FRm, @(R0,Rn)	1111nnnnmmmm0111	FRm→ (R0+Rn)	1	—	Yes	Yes	Yes
FMOV.D DRm, @(R0,Rn)	1111nnnnmmm00111	DRm→ (R0+Rn)	2	—		Yes	Yes
FMOV.S FRm, @-Rn	1111nnnnmmmm1011	Rn-=4, FRm→(Rn)	1	—	Yes	Yes	Yes
FMOV.D DRm, @-Rn	1111nnnnmmm01011	Rn-=8, DRm→(Rn)	2	—		Yes	Yes
FMOV.S FRm, @Rn	1111nnnnmmmm1010	FRm→(Rn)	1	—	Yes	Yes	Yes
FMOV.D DRm, @Rn	1111nnnnmmm01010	DRm→(Rn)	2	—		Yes	Yes
FMOV.S FRm, @(disp12,Rn)	0011nnnnmmmm000100 11dddddddddddd	FRm→(disp×4+Rn)	1	—			Yes
FMOV.D DRm, @(disp12,Rn)	0011nnnnmmm0000100 11dddddddddddd	DRm→(disp×8+Rn)	2	—			Yes
FMUL FRm, FRn	1111nnnnmmmm0010	FRn×FRm→FRn	1	—	Yes	Yes	Yes
FMUL DRm, DRn	1111nnn0mmm00010	DRn×DRm→DRn	6	—		Yes	Yes
FNEG FRn	1111nnnn01001101	-FRn→FRn	1	—	Yes	Yes	Yes
FNEG DRn	1111nnn001001101	-DRn→DRn	1	—		Yes	Yes
FSCHG	1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		Yes	Yes
FSQRT FRn	1111nnnn01101101	√FRn→FRn	9	—		Yes	Yes
FSQRT DRn	1111nnn001101101	√DRn→DRn	22	—		Yes	Yes
FSTS FPUL,FRn	1111nnnn00001101	FPUL→FRn	1	—	Yes	Yes	Yes
FSUB FRm, FRn	1111nnnnmmmm0001	FRn-FRm→FRn	1	—	Yes	Yes	Yes
FSUB DRm, DRn	1111nnn0mmm00001	DRn-DRm→DRn	6	—		Yes	Yes
FTRC FRm, FPUL	1111mmmm00111101	(long)FRm→FPUL	1	—	Yes	Yes	Yes
FTRC DRm, FPUL	1111mmm000111101	(long)DRm→FPUL	2	—		Yes	Yes

2.4.9 FPU-Related CPU Instructions

Table 2.18 FPU-Related CPU Instructions

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

2.4.10 Bit Manipulation Instructions

Table 2.19 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 → T	3	Operation result			Yes
BANDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of (disp + Rn)) & T → T	3	Ope-ration 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)) → T	3	Operation result			Yes
BLD	#imm3,Rn	10000111nnnn1iii	imm of Rn → T	1	Operation result			Yes
BLDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of (disp + Rn)) → T	3	Operation result			Yes
BOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BORNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of (disp + Rn)) | T → T	3	Operation result			Yes
BSET.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1 → (imm of (disp + Rn))	3	—			Yes
BSET	#imm3,Rn	10000110nnnn1iii	1 → imm of Rn	1	—			Yes
BST.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T → (imm of (disp + Rn))	3	—			Yes
BST	#imm3,Rn	10000111nnnn0iii	T → imm of Rn	1	—			Yes
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp + Rn)) ^ T → T	3	Operation result			Yes

2.5 Processing States

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

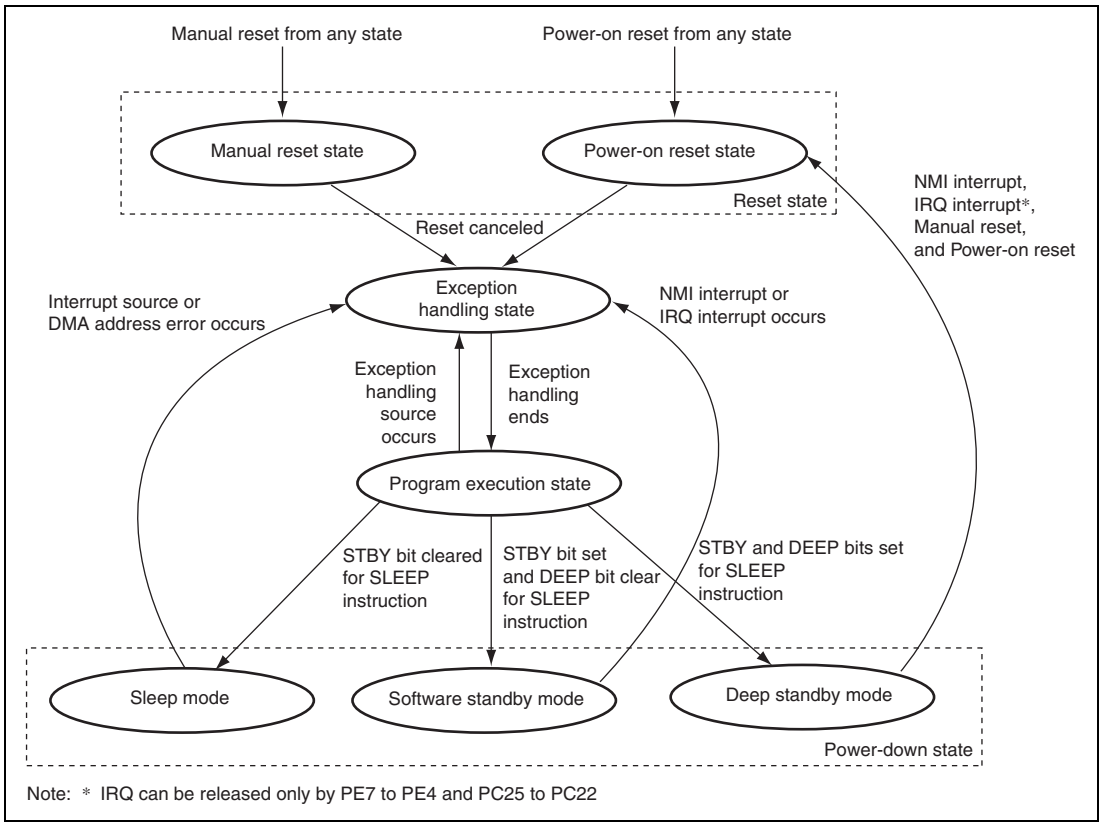


Figure 2.6 Transitions between Processing States

(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 sleep mode, software standby mode, or deep standby mode.

Section 3 Floating-Point Unit (FPU)

3.1 Features

The FPU has the following features.

- Conforms to IEEE754 standard
- 16 single-precision floating-point registers (can also be referenced as eight double-precision registers)
- Two rounding modes: Round to nearest and round to zero
- Denormalization modes: Flush to zero
- Five exception sources: Invalid operation, divide by zero, overflow, underflow, and inexact
- Comprehensive instructions: Single-precision, double-precision, and system control

3.2 Data Formats

3.2.1 Floating-Point Format

A floating-point number consists of the following three fields:

- Sign (s)
- Exponent (e)
- Fraction (f)

This LSI can handle single-precision and double-precision floating-point numbers, using the formats shown in figures 3.1 and 3.2.

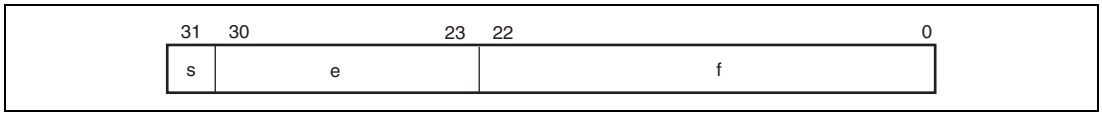


Figure 3.1 Format of Single-Precision Floating-Point Number

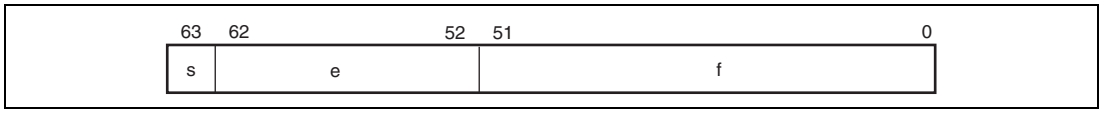


Figure 3.2 Format of Double-Precision Floating-Point Number

The exponent is expressed in biased form, as follows:

$$e = E + bias$$

The range of unbiased exponent E is $E_{min} - 1$ to $E_{max} + 1$. The two values $E_{min} - 1$ and $E_{max} + 1$ are distinguished as follows. $E_{min} - 1$ indicates zero (both positive and negative sign) and a denormalized number, and $E_{max} + 1$ indicates positive or negative infinity or a non-number (NaN). Table 3.1 shows E_{min} and E_{max} values.

Table 3.1 Floating-Point Number Formats and Parameters

Parameter	Single-Precision	Double-Precision
Total bit width	32 bits	64 bits
Sign bit	1 bit	1 bit
Exponent field	8 bits	11 bits
Fraction field	23 bits	52 bits
Precision	24 bits	53 bits
Bias	+127	+1023
E_{max}	+127	+1023
E_{min}	–126	–1022

Floating-point number value v is determined as follows:

If $E = E_{max} + 1$ and $f \neq 0$, v is a non-number (NaN) irrespective of sign s
If $E = E_{max} + 1$ and $f = 0$, $v = (-1)^s$ (infinity) [positive or negative infinity]
If $E_{min} \leq E \leq E_{max}$, $v = (-1)^s 2^E$ (1.f) [normalized number]
If $E = E_{min} - 1$ and $f \neq 0$, $v = (-1)^s 2^{Emin}$ (0.f) [denormalized number]
If $E = E_{min} - 1$ and $f = 0$, $v = (-1)^s 0$ [positive or negative zero]

Table 3.2 shows the ranges of the various numbers in hexadecimal notation.

Table 3.2 Floating-Point Ranges

Type	Single-Precision	Double-Precision
Signaling non-number	H'7FFF FFFF to H'7FC0 0000	H'7FFF FFFF FFFF FFFF to H'7FF8 0000 0000 0000
Quiet non-number	H'7FBF FFFF to H'7F80 0001	H'7FF7 FFFF FFFF FFFF to H'7FF0 0000 0000 0001
Positive infinity	H'7F80 0000	H'7FF0 0000 0000 0000
Positive normalized number	H'7F7F FFFF to H'0080 0000	H'7FEF FFFF FFFF FFFF to H'0010 0000 0000 0000
Positive denormalized number	H'007F FFFF to H'0000 0001	H'000F FFFF FFFF FFFF to H'0000 0000 0000 0001
Positive zero	H'0000 0000	H'0000 0000 0000 0000
Negative zero	H'8000 0000	H'8000 0000 0000 0000
Negative denormalized number	H'8000 0001 to H'807F FFFF	H'8000 0000 0000 0001 to H'800F FFFF FFFF FFFF
Negative normalized number	H'8080 0000 to H'FF7F FFFF	H'8010 0000 0000 0000 to H'FFEF FFFF FFFF FFFF
Negative infinity	H'FF80 0000	H'FFF0 0000 0000 0000
Quiet non-number	H'FF80 0001 to H'FFBF FFFF	H'FFF0 0000 0000 0001 to H'FFF7 FFFF FFFF FFFF
Signaling non-number	H'FFC0 0000 to H'FFFF FFFF	H'FFF8 0000 0000 0000 to H'FFFF FFFF FFFF FFFF

3.2.2 Non-Numbers (NaN)

Figure 3.3 shows the bit pattern of a non-number (NaN). A value is NaN in the following case:

- Sign bit: Don't care
- Exponent field: All bits are 1
- Fraction field: At least one bit is 1

The NaN is a signaling NaN (sNaN) if the MSB of the fraction field is 1, and a quiet NaN (qNaN) if the MSB is 0.

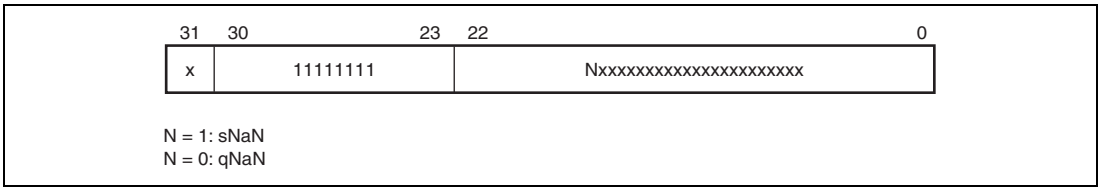


Figure 3.3 Single-Precision NaN Bit Pattern

An sNaN is input in an operation, except copy, FABS, and FNEG, that generates a floating-point value.

- When the EN.V bit in FPSCR is 0, the operation result (output) is a qNaN.
- When the EN.V bit in FPSCR is 1, an invalid operation exception will generate FPU exception processing. In this case, the contents of the operation destination register are unchanged.

If a qNaN is input in an operation that generates a floating-point value, and an sNaN has not been input in that operation, the output will always be a qNaN irrespective of the setting of the EN.V bit in FPSCR. An exception will not be generated in this case.

The qNAN values as operation results are as follows:

- Single-precision qNaN: H'7FBF FFFF
- Double-precision qNaN: H'7FF7 FFFF FFFF FFFF

See the individual instruction descriptions for details of floating-point operations when a non-number (NaN) is input.

3.2.3 Denormalized Numbers

For a denormalized number floating-point value, the exponent field is expressed as 0, and the fraction field as a non-zero value.

In the SH2A-FPU, the DN bit in the status register FPSCR is always set to 1, therefore a denormalized number (source operand or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy, FNEG, or FABS).

When the DN bit in FPSCR is 0, a denormalized number (source operand or operation result) is processed as it is. See the individual instruction descriptions for details of floating-point operations when a denormalized number is input.

3.3 Register Descriptions

3.3.1 Floating-Point Registers

Figure 3.4 shows the floating-point register configuration. There are sixteen 32-bit floating-point registers FPR0 to FPR15, referenced by specifying FR0 to FR15, DR0/2/4/6/8/10/12/14. The correspondence between FRPn and the reference name is determined by the PR and SZ bits in FPSCR. Refer figure 3.4.

1. Floating-point registers, FPRi (16 registers)
   FPR0 to FPR15
2. Single-precision floating-point registers, FRi (16 registers)
   FR0 to FR15 indicate FPR0 to FPR15
3. Double-precision floating-point registers or single-precision floating-point vector registers in pairs, DRi (8 registers)
   A DR register comprises two FR registers.
   DR0 = {FR0, FR1}, DR2 = {FR2, FR3}, DR4 = {FR4, FR5}, DR6 = {FR6, FR7},
   DR8 = {FR8, FR9}, DR10 = {FR10, FR11}, DR12 = {FR12, FR13}, DR14 = {FR14, FR15}

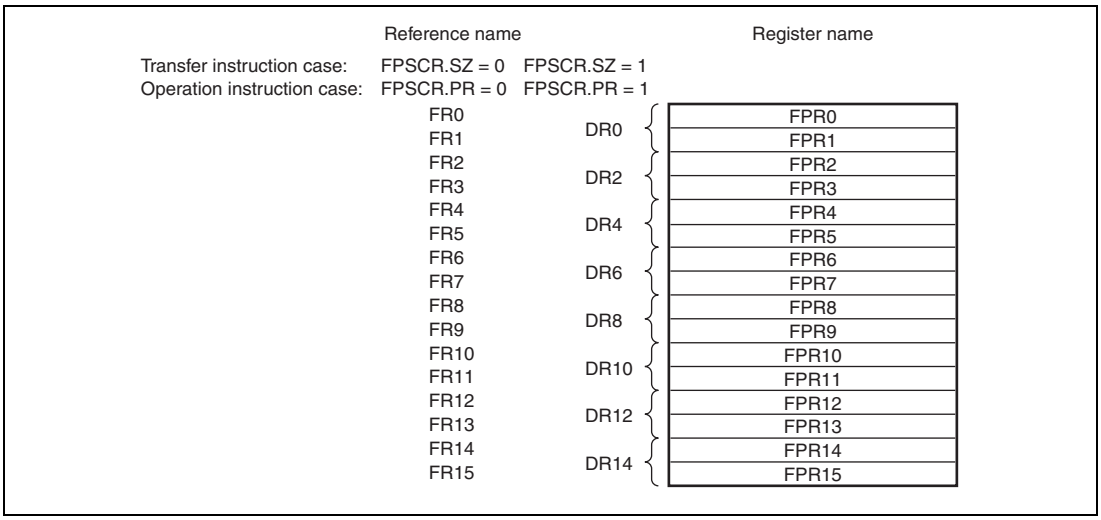


Figure 3.4 Floating-Point Registers

3.3.2 Floating-Point Status/Control Register (FPSCR)

FPSCR is a 32-bit register that controls floating-point instructions, sets FPU exceptions, and selects the rounding mode.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Cause				Enable					Flag					RM1	RM0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22	QIS	0	R/W	Nonnunerical Processing Mode 0: Processes qNaN or ±∞ as such 1: Treats qNaN or ±∞ as the same as sNaN (valid only when the V bit in FPSCR enable is set to 1)
21	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
20	SZ	0	R/W	Transfer Size Mode 0: Data size of FMOV instruction is 32-bits 1: Data size of FMOV instruction is a 32-bit register pair (64 bits)
19	PR	0	R/W	Precision Mode 0: Floating-point instructions are executed as single-precision operations 1: Floating-point instructions are executed as double-precision operations (graphics support instructions are undefined)
18	DN	1	R	Denormalization Mode (Always fixed to 1 in SH2A-FPU) 1: Denormalized number is treated as zero

Bit	Bit Name	Initial Value	R/W	Description
17 to 12	Cause	All 0	R/W	FPU Exception Cause Field FPU Exception Enable Field FPU Exception Flag Field Each time floating-point operation instruction is executed, the FPU exception cause field is cleared to 0 first. When an FPU exception on floating-point operation occurs, the bits corresponding to the FPU exception cause field and FPU exception flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. As the bits corresponding to FPU exception enable filed are sets to 1, FPU exception processing occurs. For bit allocations of each field, see table 3.3.
11 to 7	Enable	All 0	R/W	
6 to 2	Flag	All 0	R/W	
1	RM1	0	R/W	Rounding Mode These bits select the rounding mode. 00: Round to Nearest 01: Round to Zero 10: Reserved 11: Reserved
0	RM0	1	R/W	

Table 3.3 Bit Allocation for FPU Exception Handling

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

Note: No FPU error occurs in the SH2A-FPU.

3.3.3 Floating-Point Communication Register (FPUL)

Information is transferred between the FPU and CPU via FPUL. FPUL is a 32-bit system register that is accessed from the CPU side by means of LDS and STS instructions. For example, to convert the integer stored in general register R1 to a single-precision floating-point number, the processing flow is as follows:

R1 → (LDS instruction) → FPUL → (single-precision FLOAT instruction) → FR1

3.4 Rounding

In a floating-point instruction, rounding is performed when generating the final operation result from the intermediate result. Therefore, the result of combination instructions such as FMAC will differ from the result when using a basic instruction such as FADD, FSUB, or FMUL. Rounding is performed once in FMAC, but twice in FADD, FSUB, and FMUL.

Which of the two rounding methods is to be used is determined by the RM bits in FPSCR.

FPSCR.RM[1:0] = 00: Round to Nearest
FPSCR.RM[1:0] = 01: Round to Zero

(1) Round to Nearest

The operation result is rounded to the nearest expressible value. If there are two nearest expressible values, the one with an LSB of 0 is selected.

If the unrounded value is $2^{Emax} (2 - 2^{-P})$ or more, the result will be infinity with the same sign as the unrounded value. The values of Emax and P, respectively, are 127 and 24 for single-precision, and 1023 and 53 for double-precision.

(2) Round to Zero

The digits below the round bit of the unrounded value are discarded.

If the unrounded value is larger than the maximum expressible absolute value, the value will become the maximum expressible absolute value.

3.5 FPU Exceptions

3.5.1 FPU Exception Sources

FPU exceptions may occur on floating-point operation instruction and the exception sources are as follows:

- FPU error (E): When FPSCR.DN = 0 and a denormalized number is input (No chance to occur in the SH2A-FPU)
- Invalid operation (V): In case of an invalid operation, such as NaN input
- Division by zero (Z): Division with a zero divisor
- Overflow (O): When the operation result overflows
- Underflow (U): When the operation result underflows
- Inexact exception (I): When overflow, underflow, or rounding occurs

The FPU exception cause field in FPSCR contains bits corresponding to all of above sources E, V, Z, O, U, and I, and the FPU exception flag and enable fields in FPSCR contain bits corresponding to sources V, Z, O, U, and I, but not E. Thus, FPU errors cannot be disabled.

When an FPU exception occurs, the corresponding bit in the FPU exception cause field is set to 1, and 1 is added to the corresponding bit in the FPU exception flag field. When an FPU exception does not occur, the corresponding bit in the FPU exception cause field is cleared to 0, but the corresponding bit in the FPU exception flag field remains unchanged.

3.5.2 FPU Exception Handling

FPU exception handling is initiated in the following cases:

- FPU error (E): FPSCR.DN = 0 and a denormalized number is input (No chance to occur in the SH2A-FPU)
- Invalid operation (V): FPSCR.Enable.V = 1 and invalid operation
- Division by zero (Z): FPSCR.Enable.Z = 1 and division with a zero divisor
- Overflow (O): FPSCR.Enable.O = 1 and instruction with possibility of operation result overflow
- Underflow (U): FPSCR.Enable.U = 1 and instruction with possibility of operation result underflow
- Inexact exception (I): FPSCR.Enable.I = 1 and instruction with possibility of inexact operation result

These possibilities of each exceptional handling on floating-point operation are shown in the individual instruction descriptions. All exception events that originate in the floating-point operation are assigned as the same FPU exceptional handling event. The meaning of an exception generated by floating-point operation is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed when FPU exception handling operation occurs.

Except for the above, the FPU disables exception handling. In every processing, the bit corresponding to source V, Z, O, U, or I is set to 1, and a default value is generated as the operation result.

- Invalid operation (V): qNaN is generated as the result.
- Division by zero (Z): Infinity with the same sign as the unrounded value is generated.
- Overflow (O):
  When rounding mode = RZ, the maximum normalized number, with the same sign as the unrounded value, is generated.
  When rounding mode = RN, infinity with the same sign as the unrounded value is generated.
- Underflow (U):
  Zero with the same sign as the unrounded value is generated.
- Inexact exception (I): An inexact result is generated.

Section 4 Clock Pulse Generator (CPG)

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

4.1 Features

- Three clock operating modes

  The mode is selected from among the three clock operating modes by the selection of the following three conditions: the frequency-divisor in use, whether the PLLs are on or off, and whether the internal crystal resonator or the input on the external clock-signal line is used.
- Three clocks generated independently

  A CPU clock (Iϕ) for the CPU and cache; a peripheral clock (Pϕ) for the on-chip peripheral modules; a bus clock (Bϕ = CKIO) for the external bus interface.
- Frequency change function

  CPU 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 by sleep mode, software standby mode, and deep standby mode. Specific modules can also be stopped using the module standby function. For details on clock control in the power-down modes, see section 25, Power-Down Modes.

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

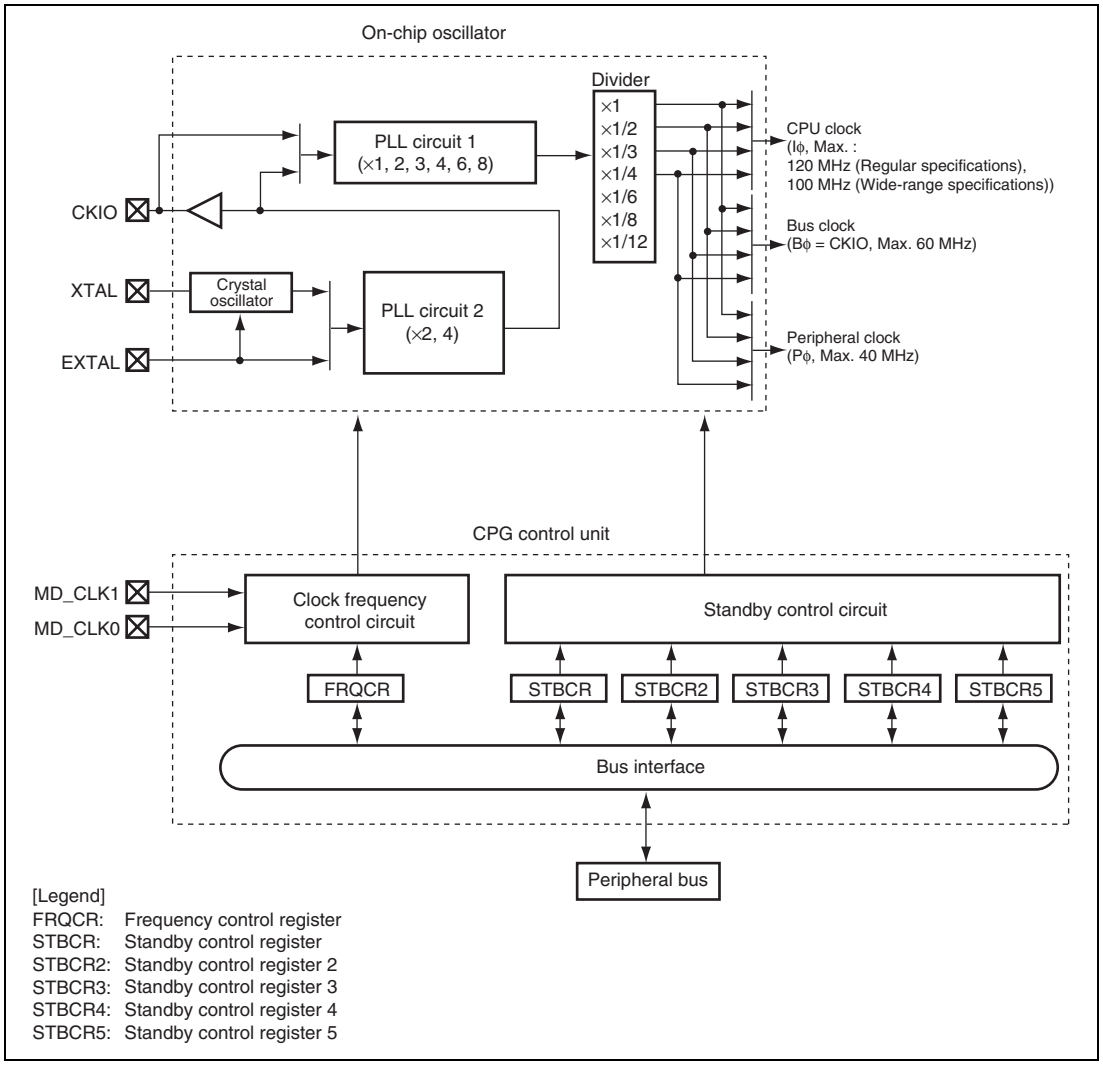


Figure 4.1 Block Diagram of Clock Pulse Generator

The clock pulse generator blocks function as follows:

(1) PLL Circuit 1

PLL circuit 1 multiplies the input clock frequency from the CKIO pin by 1, 2, 3, 4, 6, or 8. The multiplication rate is set by the frequency control register. When this is done, the phase of the rising edge of the bus clock is controlled so that it will agree with the phase of the rising edge of the CKIO pin.

(2) PLL Circuit 2

PLL circuit 2 multiplies the input clock frequency from the crystal oscillator or EXTAL pin by 2 or 4. The multiplication rate is fixed according to the clock operating mode. The clock operating mode is specified by the MD_CLK1 and MD_CLK0 pins. For details on the clock operating mode, see table 4.2.

Note that the settings of these pins cannot be changed during operation. If changed, the operation of this LSI cannot be guaranteed.

(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

Divider generates a clock signal at the operating frequency used by the CPU or peripheral clock. The operating frequency can be 1, 1/2, 1/3, 1/4, 1/6, 1/8, or 1/12 times the output frequency of PLL circuit 1, as long as it stays at or above the clock frequency of the CKIO 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_CLK1 and MD_CLK0 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 in sleep, software, and deep 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 CKIO pin during software standby mode, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the CPU clock and the peripheral clock (Pϕ).

(8) Standby Control Register

The standby control register has bits for controlling the power-down modes. See section 25, 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 Modes 0 and 2)	Function (Clock Operating Mode 3)
Mode control pins	MD_CLK0	Input	Sets the clock operating mode.	Sets the clock operating mode.
	MD_CLK1	Input	Sets the clock operating mode.	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.)	Leave this pin open.
	EXTAL	Input	Connected to the crystal resonator or used to input an external clock.	Pull-up this pin.
Clock input/output pin	CKIO	I/O	Clock output pin.	Clock input pin.

4.3 Clock Operating Modes

Table 4.2 shows the relationship between the combinations of the mode control pins (MD_CLK1 and MD_CLK0) and the clock operating modes. Table 4.2 shows the usable frequency ranges in the clock operating modes.

Table 4.2 Clock Operating Modes

	Pin Values		Clock I/O		PLL Circuit 2	PLL Circuit 1	
Mode	MD_CLK1	MD_CLK0	Source	Output	On/Off	On/Off	CKIO Frequency
0	0	0	EXTAL or crystal resonator	CKIO	ON (×4)	ON (×1, 2, 3, 4)	(EXTAL or crystal resonator) ×4
2	1	0	EXTAL or crystal resonator	CKIO	ON (×2)	ON (×1, 2, 3, 4, 6, 8)	(EXTAL or crystal resonator) ×2
3	1	1	CKIO	—	OFF	ON (×1, 2, 3, 4, 6, 8)	(CKIO)

- Mode 0

  The frequency of the signal received from the EXTAL pin or crystal resonator is quadrupled by the PLL circuit 2 before it is supplied to the LSI as the clock signal. This enables to use the external clock of lower frequency. Either a crystal resonator with a frequency in the range from 10 to 15 MHz or an external signal in the same frequency range input on the EXTAL pin may be used. The frequency range of CKIO is from 40 to 60 MHz.
- Mode 2

  The frequency of the signal received from the EXTAL pin or crystal resonator is doubled by the PLL circuit 2 before it is supplied to the LSI as the clock signal. This enables to use the external clock of lower frequency. An external signal with a frequency in the range from 10 to 30 MHz or a crystal resonator with 10 to 20 MHz may be used. The frequency range of CKIO is from 20 to 60 MHz.
- Mode 3

  In mode 3, the CKIO pin functions as an input pin and draws an external clock signal. The PLL circuit 1 shapes its waveform and the setting of the frequency control register multiplies its frequency before the clock enters the LSI. Frequency between 20 to 60 MHz can be input to the CKIO pin. For reduced current and hence power consumption, pull up the EXTAL pin and open the XTAL pin when the LSI is used in mode 3.

Table 4.3 Relationship between Clock Operating Mode and Frequency Range

Clock Operating Mode	FRQCR Setting	PLL Frequency Multiplier		Ratio of Internal Clock Frequencies (I:B:P)*1	Selectable Frequency Range (MHz)				
		PLL Circuit 1	PLL Circuit 2		Input Clock*2	Output Clock (CKIO Pin)*3	CPU Clock (Iϕ)*3	Bus Clock (Bϕ)*3	Peripheral Clock (Pϕ)*3
0	H'1000	ON (×1)	ON (×4)	4:4:4	10	40	40	40	40
	H'1001	ON (×1)	ON (×4)	4:4:2	10 to 15	40 to 60	40 to 60	40 to 60	20 to 30
	H'1002	ON (×1)	ON (×4)	4:4:4/3	10 to 15	40 to 60	40 to 60	40 to 60	13.33 to 20
	H'1003	ON (×1)	ON (×4)	4:4:1	10 to 15	40 to 60	40 to 60	40 to 60	10 to 15
	H'1004	ON (×1)	ON (×4)	4:4:2/3	10 to 15	40 to 60	40 to 60	40 to 60	6.7 to 10
	H'1005	ON (×1)	ON (×4)	4:4:1/2	10 to 15	40 to 60	40 to 60	40 to 60	5 to 7.5
	H'1006	ON (×1)	ON (×4)	4:4:1/3	10 to 15	40 to 60	40 to 60	40 to 60	3.33 to 5
	H'1101	ON (×2)	ON (×4)	8:4:4	10	40	80	40	40
	H'1103	ON (×2)	ON (×4)	8:4:2	10 to 15	40 to 60	80 to 120	40 to 60	20 to 30
	H'1104	ON (×2)	ON (×4)	8:4:4/3	10 to 15	40 to 60	80 to 120	40 to 60	13.33 to 20
	H'1105	ON (×2)	ON (×4)	8:4:1	10 to 15	40 to 60	80 to 120	40 to 60	10 to 15
	H'1106	ON (×2)	ON (×4)	8:4:2/3	10 to 15	40 to 60	80 to 120	40 to 60	6.7 to 10
	H'1111	ON (×2)	ON (×4)	4:4:4	10	40	40	40	40
	H'1113	ON (×2)	ON (×4)	4:4:2	10 to 15	40 to 60	40 to 60	40 to 60	20 to 30
	H'1114	ON (×2)	ON (×4)	4:4:4/3	10 to 15	40 to 60	40 to 60	40 to 60	13.33 to 20
	H'1115	ON (×2)	ON (×4)	4:4:1	10 to 15	40 to 60	40 to 60	40 to 60	10 to 15
	H'1116	ON (×2)	ON (×4)	4:4:2/3	10 to 15	40 to 60	40 to 60	40 to 60	6.7 to 10
	H'1202	ON (×3)	ON (×4)	4:4:4	10	40	120	40	40
	H'1204	ON (×3)	ON (×4)	4:4:2	10	40	120	40	20
	H'1206	ON (×3)	ON (×4)	4:4:1	10	40	120	40	10
	H'1222	ON (×3)	ON (×4)	4:4:4	10	40	120	40	40
	H'1224	ON (×3)	ON (×4)	4:4:2	10	40	120	40	20
	H'122C	ON (×3)	ON (×4)	4:4:2	10 to 15	40 to 60	40 to 60	40 to 60	20 to 30
	H'1226	ON (×3)	ON (×4)	4:4:1	10	40	40	40	10
	H'122E	ON (×3)	ON (×4)	4:4:1	10 to 15	40 to 60	40 to 60	40 to 60	10 to 15

Clock Operating Mode	FRQCR Setting	PLL Frequency Multiplier		Ratio of Internal Clock Frequencies (I:B:P)*1	Selectable Frequency Range (MHz)				
		PLL Circuit 1	PLL Circuit 2		Input Clock*2	Output Clock (CKIO Pin)*3	CPU Clock (Iϕ)*3	Bus Clock (Bϕ)*3	Peripheral Clock (Pϕ)*3
0	H'1313	ON (×4)	ON (×4)	8:4:4	10	40	80	40	40
	H'1315	ON (×4)	ON (×4)	8:4:2	10 to 12.5	40 to 50	80 to 100	40 to 50	20 to 25
	H'1316	ON (×4)	ON (×4)	8:4:4/3	10 to 12.5	40 to 50	80 to 100	40 to 50	13.33 to 16.67
	H'1333	ON (×4)	ON (×4)	4:4:4	10	40	40	40	40
	H'1335	ON (×4)	ON (×4)	4:4:2	10 to 12.5	40 to 50	40 to 50	40 to 50	20 to 25
	H'1336	ON (×4)	ON (×4)	4:4:4/3	10 to 12.5	40 to 50	40 to 50	40 to 50	13.33 to 16.67
2	H'1000	ON (×1)	ON (×2)	2:2:2	10 to 20	20 to 40	20 to 40	20 to 40	20 to 40
	H'1001	ON (×1)	ON (×2)	2:2:1	10 to 30	20 to 60	20 to 60	20 to 60	10 to 30
	H'1002	ON (×1)	ON (×2)	2:2:2/3	10 to 30	20 to 60	20 to 60	20 to 60	6.67 to 20
	H'1003	ON (×1)	ON (×2)	2:2:1/2	10 to 30	20 to 60	20 to 60	20 to 60	5 to 15
	H'1004	ON (×1)	ON (×2)	2:2:1/3	10 to 30	20 to 60	20 to 60	20 to 60	3.33 to 10
	H'1005	ON (×1)	ON (×2)	2:2:1/4	10 to 30	20 to 60	20 to 60	20 to 60	2.5 to 7.5
	H'1006	ON (×1)	ON (×2)	2:2:1/6	10 to 30	20 to 60	20 to 60	20 to 60	1.67 to 5
	H'1101	ON (×2)	ON (×2)	4:2:2	10 to 20	20 to 40	40 to 80	20 to 40	20 to 40
	H'1103	ON (×2)	ON (×2)	4:2:1	10 to 30	20 to 60	40 to 120	20 to 60	10 to 30
	H'1104	ON (×2)	ON (×2)	4:2:2/3	10 to 30	20 to 60	40 to 120	20 to 60	6.67 to 20
	H'1105	ON (×2)	ON (×2)	4:2:1/2	10 to 30	20 to 60	40 to 120	20 to 60	5 to 15
	H'1106	ON (×2)	ON (×2)	4:2:1/3	10 to 30	20 to 60	40 to 120	20 to 60	3.3 to 10
	H'1111	ON (×2)	ON (×2)	2:2:2	10 to 20	20 to 40	20 to 40	20 to 40	20 to 40
	H'1113	ON (×2)	ON (×2)	2:2:1	10 to 30	20 to 60	20 to 60	20 to 60	10 to 30
	H'1114	ON (×2)	ON (×2)	2:2:2/3	10 to 30	20 to 60	20 to 60	20 to 60	6.67 to 20
	H'1115	ON (×2)	ON (×2)	2:2:1/2	10 to 30	20 to 60	20 to 60	20 to 60	5 to 15
	H'1116	ON (×2)	ON (×2)	2:2:1/3	10 to 30	20 to 60	20 to 60	20 to 60	3.3 to 10
	H'1202	ON (×3)	ON (×2)	6:2:2	10 to 20	20 to 40	60 to 120	20 to 40	20 to 40
	H'120C	ON (×3)	ON (×2)	6:2:1	20	40	120	40	20
	H'120E	ON (×3)	ON (×2)	6:2:1/2	20	40	120	40	10
	H'1206	ON (×3)	ON (×2)	6:2:1/2	10 to 20	20 to 40	60 to 120	20 to 40	5 to 10
	H'1222	ON (×3)	ON (×2)	2:2:2	10 to 20	20 to 40	20 to 40	20 to 40	20 to 40
	H'1224	ON (×3)	ON (×2)	2:2:1	10 to 20	20 to 40	20 to 40	20 to 40	10 to 20
	H'122C	ON (×3)	ON (×2)	2:2:1	20 to 30	40 to 60	40 to 60	40 to 60	20 to 30

Clock Operating Mode	FRQCR Setting	PLL Frequency Multiplier		Ratio of Internal Clock Frequencies (I:B:P)*1	Selectable Frequency Range (MHz)				
		PLL Circuit 1	PLL Circuit 2		Input Clock*2	Output Clock (CKIO Pin)*3	CPU Clock (Iϕ)*3	Bus Clock (Bϕ)*3	Peripheral Clock (Pϕ)*3
2	H'1226	ON (×3)	ON (×2)	2:2:1/2	10 to 20	20 to 40	20 to 40	20 to 40	5 to 10
	H'1303	ON (×4)	ON (×2)	8:2:2	10 to 15	20 to 30	80 to 120	20 to 30	20 to 30
	H'1305	ON (×4)	ON (×2)	8:2:1	10 to 15	20 to 30	80 to 120	20 to 30	10 to 15
	H'1306	ON (×4)	ON (×2)	8:2:2/3	10 to 15	20 to 30	80 to 120	20 to 30	6.67 to 10
	H'1313	ON (×4)	ON (×2)	4:2:2	10 to 20	20 to 40	40 to 80	20 to 40	20 to 40
	H'1315	ON (×4)	ON (×2)	4:2:1	10 to 25	20 to 50	40 to 100	20 to 50	10 to 25
	H'1316	ON (×4)	ON (×2)	4:2:2/3	10 to 25	20 to 50	40 to 100	20 to 50	6.67 to 16.67
	H'1333	ON (×4)	ON (×2)	2:2:2	10 to 20	20 to 40	20 to 40	20 to 40	20 to 40
	H'1335	ON (×4)	ON (×2)	2:2:1	10 to 25	20 to 50	20 to 50	20 to 50	10 to 25
	H'1336	ON (×4)	ON (×2)	2:2:2/3	10 to 25	20 to 50	20 to 50	20 to 50	6.67 to 16.67
	H'1404	ON (×6)	ON (×2)	12:2:2	10	20	120	20	20
	H'1406	ON (×6)	ON (×2)	12:2:1	10	20	120	20	10
	H'1414	ON (×6)	ON (×2)	6:2:2	10 to 16.67	20 to 33.33	60 to 100	20 to 33.33	20 to 33.33
	H'1416	ON (×6)	ON (×2)	6:2:1	10 to 16.67	20 to 33.33	60 to 100	20 to 33.33	10 to 16.67
	H'1424	ON (×6)	ON (×2)	4:2:2	10 to 16.67	20 to 33.33	40 to 66.67	20 to 33.33	20 to 33.33
	H'1426	ON (×6)	ON (×2)	4:2:1	10 to 16.67	20 to 33.33	40 to 66.67	20 to 33.33	10 to 16.67
	H'1444	ON (×6)	ON (×2)	2:2:2	10 to 16.67	20 to 33.33	20 to 33.33	20 to 33.33	20 to 33.33
	H'1446	ON (×6)	ON (×2)	2:2:1	10 to 16.67	20 to 33.33	20 to 33.33	20 to 33.33	10 to 16.67
	H'1515	ON (×8)	ON (×2)	8:2:2	10 to 12.5	20 to 25	80 to 100	20 to 25	20 to 25
	H'1535	ON (×8)	ON (×2)	4:2:2	10 to 12.5	20 to 25	40 to 50	20 to 25	20 to 25
	H'1555	ON (×8)	ON (×2)	2:2:2	10 to 12.5	20 to 25	20 to 25	20 to 25	20 to 25
3	H'1000	ON (×1)	OFF	1:1:1	20 to 40	—	20 to 40	20 to 40	20 to 40
	H'1001	ON (×1)	OFF	1:1:1/2	20 to 60	—	20 to 60	20 to 60	10 to 30
	H'1002	ON (×1)	OFF	1:1:1/3	20 to 60	—	20 to 60	20 to 60	6.67 to 20
	H'1003	ON (×1)	OFF	1:1:1/4	20 to 60	—	20 to 60	20 to 60	5 to 15
	H'1004	ON (×1)	OFF	1:1:1/6	20 to 60	—	20 to 60	20 to 60	3.33 to 10
	H'1005	ON (×1)	OFF	1:1:1/8	20 to 60	—	20 to 60	20 to 60	2.5 to 7.5
	H'1006	ON (×1)	OFF	1:1:1/12	20 to 60	—	20 to 60	20 to 60	1.67 to 5

Clock Operating Mode	FRQCR Setting	PLL Frequency Multiplier		Ratio of Internal Clock Frequencies (I:B:P)*1	Selectable Frequency Range (MHz)				
		PLL Circuit 1	PLL Circuit 2		Input Clock*2	Output Clock (CKIO Pin)*3	CPU Clock (Iφ)*3	Bus Clock (Bφ)*3	Peripheral Clock (Pφ)*3
3	H'1101	ON (×2)	OFF	2:1:1	20 to 40	—	40 to 80	20 to 40	20 to 40
	H'1103	ON (×2)	OFF	2:1:1/2	20 to 60	—	40 to 120	20 to 60	10 to 30
	H'1104	ON (×2)	OFF	2:1:1/3	20 to 60	—	40 to 120	20 to 60	6.67 to 20
	H'1105	ON (×2)	OFF	2:1:1/4	20 to 60	—	40 to 120	20 to 60	5 to 15
	H'1106	ON (×2)	OFF	2:1:1/6	20 to 60	—	40 to 120	20 to 60	3.33 to 10
	H'1111	ON (×2)	OFF	1:1:1	20 to 40	—	20 to 40	20 to 40	20 to 40
	H'1113	ON (×2)	OFF	1:1:1/2	20 to 60	—	20 to 60	20 to 60	10 to 30
	H'1114	ON (×2)	OFF	1:1:1/3	20 to 60	—	20 to 60	20 to 60	6.67 to 20
	H'1115	ON (×2)	OFF	1:1:1/4	20 to 60	—	20 to 60	20 to 60	5 to 15
	H'1116	ON (×2)	OFF	1:1:1/6	20 to 60	—	20 to 60	20 to 60	3.33 to 10
	H'1202	ON (×3)	OFF	3:1:1	20 to 40	—	60 to 120	20 to 40	20 to 40
	H'120C	ON (×3)	OFF	3:1:1/2	40	—	120	40	20
	H'1206	ON (×3)	OFF	3:1:1/4	20 to 40	—	60 to 120	20 to 40	5 to 10
	H'1222	ON (×3)	OFF	1:1:1	20 to 40	—	20 to 40	20 to 40	20 to 40
	H'1224	ON (×3)	OFF	1:1:1/2	20 to 40	—	20 to 40	20 to 40	10 to 20
	H'122C	ON (×3)	OFF	1:1:1/2	20 to 60	—	40 to 60	40 to 60	20 to 30
	H'1226	ON (×3)	OFF	1:1:1/4	20 to 40	—	20 to 40	20 to 40	5 to 10
	H'122E	ON (×3)	OFF	1:1:1/4	40 to 60	—	40 to 60	40 to 60	10 to 15
	H'1303	ON (×4)	OFF	4:1:1	20 to 30	—	80 to 120	20 to 30	20 to 30
	H'1305	ON (×4)	OFF	4:1:1/2	20 to 30	—	80 to 120	20 to 30	10 to 15
	H'1306	ON (×4)	OFF	4:1:1/3	20 to 30	—	80 to 120	20 to 30	6.67 to 10
	H'1313	ON (×4)	OFF	2:1:1	20 to 40	—	40 to 80	20 to 40	20 to 40
	H'1315	ON (×4)	OFF	2:1:1/2	20 to 50	—	40 to 100	20 to 50	10 to 25
	H'1316	ON (×4)	OFF	2:1:1/3	20 to 50	—	40 to 100	20 to 50	6.67 to 16.67
	H'1333	ON (×4)	OFF	1:1:1	20 to 40	—	20 to 40	20 to 40	20 to 40
	H'1335	ON (×4)	OFF	1:1:1/2	20 to 50	—	20 to 50	20 to 50	10 to 25
	H'1336	ON (×4)	OFF	1:1:1/3	20 to 50	—	20 to 50	20 to 50	6.67 to 16.67

Clock Operating Mode	FRQCR Setting	PLL Frequency Multiplier		Ratio of Internal Clock Frequencies (I:B:P)*1	Selectable Frequency Range (MHz)				
		PLL Circuit 1	PLL Circuit 2		Input Clock*2	Output Clock (CKIO Pin)*3	CPU Clock (Iϕ)*3	Bus Clock (Bϕ)*3	Peripheral Clock (Pϕ)*3
3	H'1404	ON (×6)	OFF	6:1:1	20	—	120	20	20
	H'1406	ON (×6)	OFF	6:1:1/2	20	—	120	20	10
	H'1414	ON (×6)	OFF	3:1:1	20 to 33.33	—	60 to 100	20 to 33.33	20 to 33.33
	H'1416	ON (×6)	OFF	3:1:1/2	20 to 33.33	—	60 to 100	20 to 33.33	10 to 16.67
	H'1424	ON (×6)	OFF	2:1:1	20 to 33.33	—	40 to 66.67	20 to 33.33	20 to 33.33
	H'1426	ON (×6)	OFF	2:1:1/2	20 to 33.33	—	40 to 66.67	20 to 33.33	10 to 16.67
	H'1444	ON (×6)	OFF	1:1:1	20 to 33.33	—	20 to 33.33	20 to 33.33	20 to 33.33
	H'1446	ON (×6)	OFF	1:1:1/2	20 to 33.33	—	20 to 33.33	20 to 33.33	10 to 16.67
	H'1515	ON (×8)	OFF	4:1:1	20 to 25	—	80 to 100	20 to 25	20 to 25
	H'1535	ON (×8)	OFF	2:1:1	20 to 25	—	40 to 50	20 to 25	20 to 25
	H'1555	ON (×8)	OFF	1:1:1	20 to 25	—	20 to 25	20 to 25	20 to 25

Notes:
1. The ratio of clock frequencies, where the input clock frequency is assumed to be 1.
2. In modes 0 and 2, the frequency of the clock input from the EXTAL pin or the frequency of the crystal resonator. In mode 3, the frequency of the clock input from the CKIO pin.
3. Use an internal clock (Iϕ) frequency of 120 MHz or lower for the regular specifications and 100 MHz or lower for the wide-range specifications. Use a CKIO pin or bus clock (Bϕ) frequency of 60 MHz or lower. Pϕ must be from 5 through 40 MHz.

Caution: Do not use this LSI for frequency settings other than those in table 4.3.

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
CKIO control register	CKIOCR	R/W	H'10/H'00	H'FFFE3894	8, 16, 32

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 CKIO pin in software standby mode, the frequency multiplication ratio of PLL circuit 1, and the frequency division ratio of the CPU clock and peripheral clock (Pϕ). Only word access can be used on FRQCR.

FRQCR is initialized to H'1003 only by a power-on reset or in deep standby mode. 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[2:0]			—	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/W	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 from the CKIO pin, or whether the CKIO pin is placed in the level-fixed state during software standby mode or cancellation of software standby mode. If this bit is cleared to 0, the CKIO pin is fixed at low during software standby mode or cancellation of software standby mode. Therefore, the malfunction of an external circuit because of an unstable CKIO clock during cancellation of software standby mode can be prevented. In clock operating mode 3, the CKIO pin functions as an input regardless of this bit value. 0: The CKIO pin is fixed to the low level during software standby mode or cancellation of software standby mode. 1: Clock is output from CKIO pin (low level in software standby mode).
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	STC[2:0]	000	R/W	Frequency Multiplication Ratio of PLL Circuit 1 000: × 1 time 001: × 2 times 010: × 3 times 011: × 4 times 100: × 6 times 101: × 8 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	CPU Clock Frequency Division Ratio These bits specify the frequency division ratio of the CPU clock with respect to the output frequency of PLL circuit 1. 000: × 1 time 001: × 1/2 time 010: × 1/3 time 011: × 1/4 time 100: × 1/6 time 101: × 1/8 time
3	RNGS	0	R/W	Output Range Select for PLL Circuit 1 When the multiplication ratio for the PLL circuit 1 is specified to × 3, set this bit according to the output frequency of the PLL circuit 1. 0: Low frequency mode (Output frequency of the PLL circuit 1 is equal to or smaller than 120 MHz.) 1: High frequency mode (Multiplication ratio for the PLL circuit 1 is specified to × 3 and its output frequency exceeds 120 MHz.)
2 to 0	PFC[2:0]	011	R/W	Peripheral Clock Frequency Division Ratio These bits specify the frequency division ratio of the peripheral clock with respect to the output frequency of PLL circuit 1. 000: × 1 time 001: × 1/2 time 010: × 1/3 time 011: × 1/4 time 100: × 1/6 time 101: × 1/8 time 110: × 1/12 time

4.4.2 CKIO Control Register (CKIOCR)

CKIOCR is an 8-bit readable/writable register used to control output of the CKIO pin. When this LSI is started in clock operating mode 3, writing 1 to this register is invalid.

When this LSI is started in clock operating mode 3, CKIOCR is initialized to H'00 by a power-on reset caused by the $\overline{\text{RES}}$ pin or in deep standby mode. When this LSI is started in clock operating mode 0 or 2, CKIOCR is initialized to H'01 by a power-on reset caused by the $\overline{\text{RES}}$ pin or in deep standby mode. This register is not initialized by an internal reset triggered by an overflow of the WDT, a manual reset, in sleep mode, or in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CKIO OE
Initial value:	0	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	CKIOOE	0/1*	R/W	CKIO Output Enable Enables output of the CKIO pin. 0: Output from CKIO is not enabled. 1: Output from CKIO is enabled.

Note: * The initial value depends on the clock operating mode of the LSI.

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. The on-chip WDT counts the settling time.

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
3. Set the desired value in the STC[2:0] 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 CKIO pin. This state is the same as software standby mode. Whether or not registers are initialized depends on the module. For details, see section 25, 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 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.

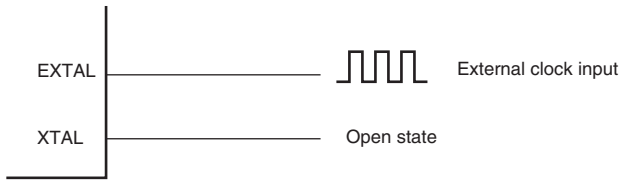
1. In the initial state, IFC[2:0] = B'000 and PFC[2:0] = B'011.
2. Set the desired value in the IFC[2:0] and PFC[2:0] bits. The values that can be set are limited by the clock operating mode and the multiplication rate of PLL circuit 1. Note that if the wrong value is set, this LSI will malfunction.
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 Inputting External Clock

Figure 4.2 is an example of connecting the external clock input. When putting the XTAL pin in open state, make sure the parasitic capacitance is less than or equal to 10 pF. To stably input the external clock with enough PLL stabilizing time at power on or releasing the standby, wait longer than the oscillation stabilizing time.



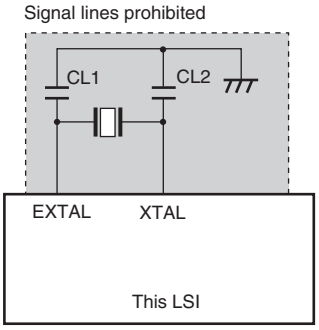
Example of connection with XTAL pin open

Figure 4.2 Example of Connecting External Clock

For details on input conditions of the external clock, see section 29.3.1, Clock Timing.

4.6.2 Note on Using Crystal Resonator

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



Reference value
CL1 = 10 to 22 pF
CL2 = 10 to 22 pF

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

Figure 4.3 Note on Using Crystal Resonator

4.6.3 Note on Resonator

Since various characteristics related to the resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, using the resonator connection examples shown in this section as a guide. As the parameters for the oscillation circuit will depend on the floating capacitance of the resonator and the user board, the parameters should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the resonator pin.

4.6.4 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.

In clock operating mode 3, the EXTAL pin is pulled up and the XTAL pin is left open.

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 the digital power supply pins VccR and PVcc should not supply the same resources on the board if at all possible.

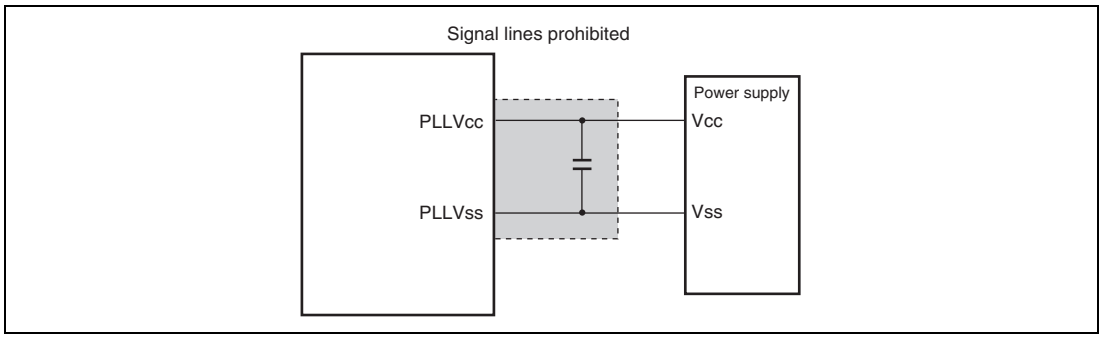


Figure 4.4 Note on Using PLL Oscillation Circuit

4.6.5 Note on Changing the Multiplication Rate

If the multiplication rate is changed by the frequency control register (FRQCR) during transfer by the DMAC, the DMAC stops its operation without waiting for the completion of the transfer. Thus, the DMA transfer is not guaranteed. Therefore, when changing the multiplication rate with the frequency control register (FRQCR), wait for the completion of the DMA transfer or stop the DMA transfer to change the setting of the frequency control register (FRQCR).

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, bus 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	
Bus error	Bus error	
Instructions	FPU exception	
	Integer division exception (division by zero)	
	Integer division exception (overflow)	
Register bank error	Bank underflow	
	Bank overflow	
Interrupts	NMI	
	User break	
	H-UDI	
	IRQ	
	PINT	Low

Type	Exception Handling		Priority
Interrupts	On-chip peripheral modules	A/D converter (ADC)	High
		Multifunction timer pulse unit 2 (MTU2)	
		Realtime clock (RTC)	
		Watchdog timer (WDT)	
		I²C bus interface 3 (IIC3)	
		Direct memory access controller (DMAC)	
		Serial communication interface with FIFO (SCIF)	
		Controller area network (RCAN-ET)	
		Serial sound interface (SSI)	
		8-bit timer (TMR)	
Instructions	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.
Bus error		
Interrupts		
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 exceptions	Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by −1.
	FPU exceptions	Exception handling starts triggered by disabled operation exception of floating-point operation instruction (IEEE754 standard), division exception by zero, overflow, underflow, or imprecise exception. Setting the QIS bit in FPSCR or inputting qNaN as well as ±∞ as the floating-point operation instruction source also starts exception handling.

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. FPSCR is initialized to H'00040001 by a power-on reset. The program begins running from the PC address fetched from the exception handling vector table.

(2) Exception Handling Triggered by Address Errors, Bus 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 the NMI or user break, 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, bus error, register bank error, NMI interrupt, user break 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
Bus error		10	H'00000028 to H'0000002B
Interrupts	NMI	11	H'0000002C to H'0000002F
	User break	12	H'00000030 to H'00000033
FPU exception		13	H'00000034 to H'00000037
H-UDI		14	H'00000038 to H'0000003B
Bank overflow		15	H'0000003C to H'0000003F
Bank underflow		16	H'00000040 to H'00000043
Integer division exception (division by zero)		17	H'00000044 to H'00000047
Integer division exception (overflow)		18	H'00000048 to H'0000004B

Exception Sources	Vector Numbers	Vector Table Address Offset
(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, PINT), on-chip peripheral module interrupts*	64	H'00000100 to H'00000103
	:	:
	255	H'000003FC to H'000003FF

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, bus 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 Input/Output Pins

Table 5.5 shows the configuration of pins relating to the resets.

Table 5.5 Pin Configuration

Pin Name	Symbol	I/O	Function
Power-on reset	$\overline{\text{RES}}$	Input	When this pin is driven low, this LSI shifts to the power-on reset processing
Manual reset	$\overline{\text{MRES}}$	Input	When this pin is driven low, this LSI shifts to the manual reset processing.

5.2.2 Types of Reset

A reset is the highest-priority exception handling source. There are two kinds of resets, power-on and manual. As shown in table 5.6, the CPU state is initialized by both a power-on reset and a manual reset. The FPU state is initialized by a power-on reset, but not by a manual reset. On-chip peripheral module registers except a few registers are initialized by a power-on reset, but not by a manual reset.

Table 5.6 Reset States

	Conditions for Transition to Reset State				Internal States		
Type	$\overline{\text{RES}}$	H-UDI Command	$\overline{\text{MRES}}$	WDT Overflow	CPU	On-Chip Peripheral Modules, I/O Port	WRCSR of WDT, FRQCR of CPG
Power-on reset	Low	—	—	—	Initialized	Initialized*1	Initialized
	High	H-UDI reset assert command is set	—	—	Initialized	Initialized*1	Initialized
	High	Command other than H-UDI reset assert is set	—	Power-on reset	Initialized	Initialized*1	Not initialized

	Conditions for Transition to Reset State				Internal States		
Type	$\overline{\text{RES}}$	H-UDI Command	$\overline{\text{MRES}}$	WDT Overflow	CPU	On-Chip Peripheral Modules, I/O Port	WRCSR of WDT, FRQCR of CPG
Manual reset	High	Command other than H-UDI reset assert is set	Low	—	Initialized	Not initialized*2	Not initialized
	High	Command other than H-UDI reset assert is set	High	Manual reset	Initialized	Not initialized*2	Not initialized

Notes: 1. Some registers are excluded. For details, see section 28.3, Register States in Each Operating Mode.
2. The BN bit in IBNR of the INTC is initialized.

5.2.3 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 to 0. The BN bit in IBNR of the INTC is also initialized to 0. FPSCR is initialized to H'00040001.
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.

5.2.4 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.

(3) Notes at a Manual Reset

When a manual reset is generated, the bus cycle is retained. Thus, manual reset exception handling will be deferred until the CPU acquires the bus mastership. The CPU and the BN bit in IBNR of the INTC are initialized by a manual reset. The FPU and other modules are not initialized.

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 5.7.

Table 5.7 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 area other than H'F0000000 to H'F5FFFFFF in cache address array space*[1]	None (normal)
		Instruction fetched from H'F0000000 to H'F5FFFFFF in cache address array space*[1]	Address error occurs
Data read/write	CPU	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
		Double longword data accessed from double longword boundary	None (normal)
		Double longword data accessed from other than double longword boundary	Address error occurs
		Byte or word data accessed in on-chip peripheral module space*[2]	None (normal)
		Longword data accessed in 16-bit on-chip peripheral module space*[2]	None (normal)
		Longword data accessed in 8-bit on-chip peripheral module space*[2]	None (normal)

Notes: 1. For details on cache address array space, see section 8, Cache.
2. For details on peripheral module space, see section 9, Bus State Controller (BSC).

5.3.2 Address Error Exception Handling

When an address error occurs, address error exception handling starts after the bus cycle in which the address error occurred ends* and execution of the instruction being executed completes. The CPU operates as follows.

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

Note: * In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred doesn't end until the entire three above-mentioned operations end, the CPU will start address error exception handling again until the bus cycle in which the address error occurred ends.

5.4 Bus Error

5.4.1 Bus Error Generation Source

In bus monitor, notification of bus error occurrence to the CPU can be set. The notification is generated when incorrect address access or bus timeout is detected. For details, see section 10, Bus Monitor.

5.4.2 Bus Error Exception Handling

When a bus error occurs, bus error exception handling starts after the bus cycle in which the bus error occurred ends and execution of the instruction being executed completes. The CPU operates as follows.

1. The exception service routine start address which corresponds to the bus 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.

5.5 Register Bank Errors

5.5.1 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.

5.5.2 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.

5.6 Interrupts

5.6.1 Interrupt Sources

Table 5.8 shows the sources that start up interrupt exception handling. These are divided into NMI, user breaks, H-UDI, IRQ, PINT, and on-chip peripheral modules.

Table 5.8 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
PINT	PINT0 to PINT7 pins (external input)	8
On-chip peripheral module	A/D converter (ADC)	1
	Multifunction timer pulse unit 2 (MTU2)	28
	Realtime clock (RTC)	3
	Watchdog timer (WDT)	1
	I²C bus interface 3 (IIC3)	15
	Direct memory access controller (DMAC)	9
	Serial communication interface with FIFO (SCIF)	32
	Controller area network (RCAN-ET)	2
	Serial sound interface (SSI)	2
	8-bit timer (TMR)	6

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.

5.6.2 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, PINT interrupts, and on-chip peripheral module interrupts can be set freely using the interrupt priority registers 01, 02, and 05 to 16 (IPR01, IPR02, and IPR05 to IPR16) of the INTC as shown in table 5.9. 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 16 (IPR01, IPR02, IPR05 to IPR16), for details of IPR01, IPR02, and IPR05 to IPR16.

Table 5.9 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 16 (IPR01, IPR02, and IPR05 to IPR16)
PINT		
On-chip peripheral module	0 to 15	Set with interrupt priority registers 01, 02, and 05 to 16 (IPR01, IPR02, and IPR05 to IPR16)

5.6.3 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 the NMI or user break, 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, bus error, NMI interrupt, user break 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 the interrupt controller is set to accept register bank overflow exceptions (the BOVE bit in IBNR of INTC is set to 1), a register bank overflow exception will occur. 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 of the interrupt exception service routine 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.

5.7 Exceptions Triggered by Instructions

5.7.1 Types of Exceptions Triggered by Instructions

Exception handling can be triggered by trap instructions, general illegal instructions, slot illegal instructions, integer division exceptions, and FPU exceptions, as shown in table 5.10.

Table 5.10 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
FPU exceptions	Instructions that cause disabled operation exception defined by IEEE754 standard or division exception by zero. Instructions that could cause overflow, underflow, or imprecise exception.	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT

5.7.2 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.

5.7.3 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.

5.7.4 General Illegal Instructions

When undefined code placed anywhere other than immediately after a delayed branch instruction (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.

5.7.5 Integer Division Exceptions

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 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.

5.7.6 FPU Exceptions

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

The floating-point operation instructions that may cause generation of an FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

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

1. The start address of the exception service routine which corresponds to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved to the stack.
3. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.

4. After jumping to the address fetched from the exception handling vector table, program execution starts. The jump that occurs is not a delayed branch.

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

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

5.8 When Exception Sources Are Not Accepted

When an address error, bus error, FPU exception, 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 5.11. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

Table 5.11 Exception Source Generation Immediately after Delayed Branch Instruction

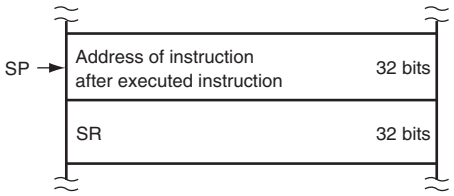
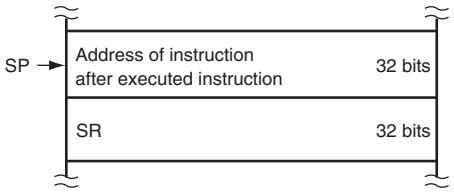
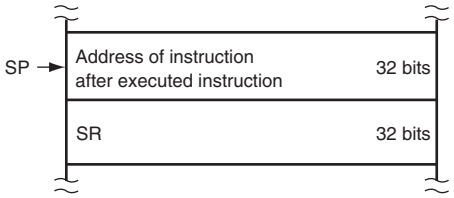
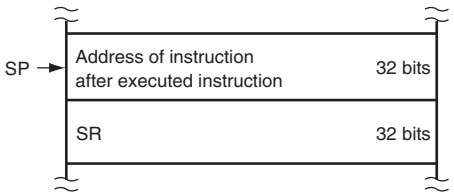
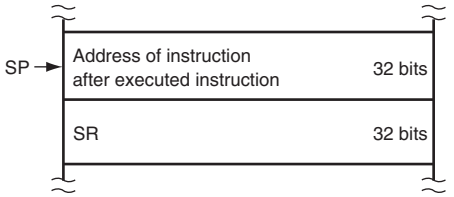
	Exception Source				
Point of Occurrence	**Address Error**	**Bus Error**	**FPU Exception**	**Register Bank Error (Overflow)**	**Interrupt**
Immediately after a delayed branch instruction*	Not accepted	Not accepted	Not accepted	Not accepted	Not accepted

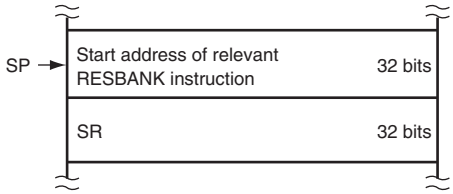
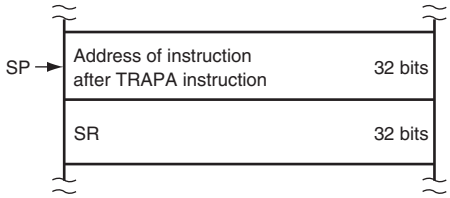
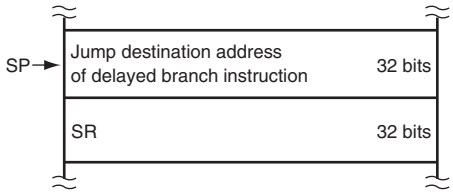
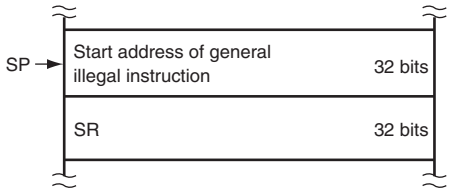
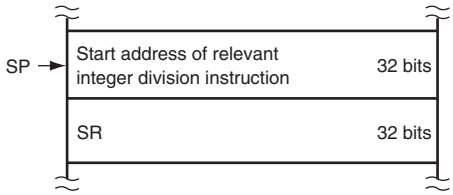
Note: * Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.

5.9 Stack Status after Exception Handling Ends

The status of the stack after exception handling ends is as shown in table 5.12.

Table 5.12 Stack Status After Exception Handling Ends

Exception Type	Stack Status
Address error	
Interrupt	
Bus error	
FPU exception	
Register bank error (overflow)	

Exception Type	Stack Status
Register bank error (underflow)	
Trap instruction	
Slot illegal instruction	
General illegal instruction	
Integer division exception	

5.10 Usage Notes

5.10.1 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.

5.10.2 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.

5.10.3 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 the 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 14 interrupt priority registers, the priorities of the IRQ, PINT, and on-chip peripheral module interrupts can be set to one of 16 levels for each source.
- NMI noise canceller function

  This controller provides an NMI input level bit that indicates the NMI pin state. The interrupt exception service routine can verify the pin state by reading this bit and use the information to implement a noise canceling function.
- Register banks

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

Figure 6.1 shows a block diagram of the INTC.

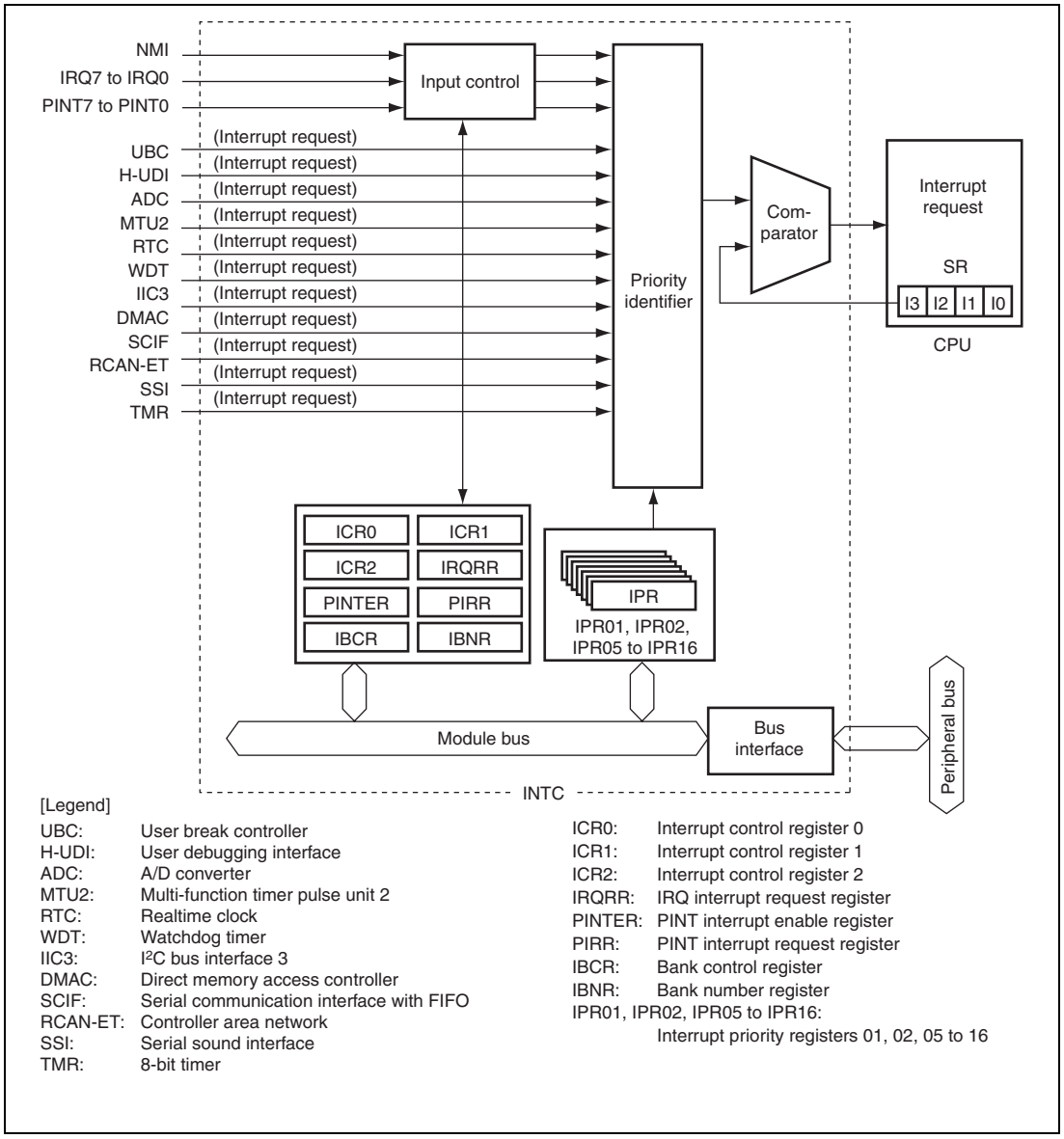


Figure 6.1 Block Diagram of INTC

6.2 Input/Output Pins

Table 6.1 shows the pin configuration of the INTC.

Table 6.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Nonmaskable interrupt input pin	NMI	Input	Input of nonmaskable interrupt request signal
Interrupt request input pins	IRQ7 to IRQ0	Input	Input of maskable interrupt request signals
	PINT7 to PINT0	Input	

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'FFFD9400	16, 32
Interrupt control register 1	ICR1	R/W	H'0000	H'FFFD9402	16
Interrupt control register 2	ICR2	R/W	H'0000	H'FFFD9404	16, 32
IRQ interrupt request register	IRQRR	R/(W)*2	H'0000	H'FFFD9406	16
PINT interrupt enable register	PINTER	R/W	H'0000	H'FFFD9408	16, 32
PINT interrupt request register	PIRR	R	H'0000	H'FFFD940A	16
Bank control register	IBCR	R/W	H'0000	H'FFFD940C	16, 32
Bank number register	IBNR	R/W	H'0000	H'FFFD940E	16
Interrupt priority register 01	IPR01	R/W	H'0000	H'FFFD9418	16, 32
Interrupt priority register 02	IPR02	R/W	H'0000	H'FFFD941A	16
Interrupt priority register 05	IPR05	R/W	H'0000	H'FFFD9420	16
Interrupt priority register 06	IPR06	R/W	H'0000	H'FFFD9800	16, 32
Interrupt priority register 07	IPR07	R/W	H'0000	H'FFFD9802	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Interrupt priority register 08	IPR08	R/W	H'0000	H'FFFD9804	16, 32
Interrupt priority register 09	IPR09	R/W	H'0000	H'FFFD9806	16
Interrupt priority register 10	IPR10	R/W	H'0000	H'FFFD9808	16, 32
Interrupt priority register 11	IPR11	R/W	H'0000	H'FFFD980A	16
Interrupt priority register 12	IPR12	R/W	H'0000	H'FFFD980C	16, 32
Interrupt priority register 13	IPR13	R/W	H'0000	H'FFFD980E	16
Interrupt priority register 14	IPR14	R/W	H'0000	H'FFFD9810	16, 32
Interrupt priority register 15	IPR15	R/W	H'0000	H'FFFD9812	16
Interrupt priority register 16	IPR16	R/W	H'0000	H'FFFD9814	16
DMA transfer request enable register 0	DREQER0	R/W	H'00	H'FFFF1600	8, 16, 32
DMA transfer request enable register 1	DREQER1	R/W	H'00	H'FFFF1601	8
DMA transfer request enable register 2	DREQER2	R/W	H'00	H'FFFF1602	8, 16
DMA transfer request enable register 3	DREQER3	R/W	H'00	H'FFFF1603	8

Notes: 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 16 (IPR01, IPR02, IPR05 to IPR16)

IPR01, IPR02, and IPR05 to IPR16 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 IPR16.

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 IPR16

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	PINT0 to PINT7	Reserved	ADI	Reserved
Interrupt priority register 06	Reserved	MTU0 (TGI0A to TGI0D)	MTU0 (TCI0V, TGI0E, TGI0F)	MTU1 (TGI1A, TGI1B)
Interrupt priority register 07	MTU1 (TCI1V, TCI1U)	MTU2 (TGI2A, TGI2B)	MTU2 (TCI2V, TCI2U)	MTU3 (TGI3A to TGI3D)
Interrupt priority register 08	MTU3 (TGI3V)	MTU4 (TGI4A to TGI4D)	MTU4 (TGI4V)	MTU5 (TGI5U, TGI5V, TGI5W)
Interrupt priority register 09	RTC	WDT	IIC0	Reserved
Interrupt priority register 10	IIC1	IIC2	DMAC0	DMAC1
Interrupt priority register 11	DMAC2	DMAC3	SCIF0	SCIF1
Interrupt priority register 12	SCIF2	SCIF3	SCIF4	SCIF5

Register Name	Bits 15 to 12	Bits 11 to 8	Bits 7 to 4	Bits 3 to 0
Interrupt priority register 13	SCIF6	SCIF7	DMINTA	DMAC4
Interrupt priority register 14	DMAC5	DMAC6	DMAC7	Reserved
Interrupt priority register 15	Reserved	RCAN-ET0	RCAN-ET1	Reserved
Interrupt priority register 16	SSI0	SSI1	TMR0	TMR1

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 IPR16 are initialized to H'0000 by a power-on reset or in deep standby mode.

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 or in deep standby mode.

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 or in deep standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ7 1S	IRQ7 0S	IRQ6 1S	IRQ6 0S	IRQ5 1S	IRQ5 0S	IRQ4 1S	IRQ4 0S	IRQ3 1S	IRQ3 0S	IRQ2 1S	IRQ2 0S	IRQ1 1S	IRQ1 0S	IRQ0 1S	IRQ0 0S
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 Interrupt Control Register 2 (ICR2)

ICR2 is a 16-bit register that specifies the detection mode for external interrupt input pins PINT7 to PINT0 individually: low level or high level. ICR2 is initialized by a power-on reset or in deep standby mode.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PINT7S	0	R/W	PINT Sense Select These bits select whether interrupt signals corresponding to pins PINT7 to PINT0 are detected by a low level or high level. 0: Interrupt request is detected on low level of PINTn input 1: Interrupt request is detected on high level of PINTn input
6	PINT6S	0	R/W	
5	PINT5S	0	R/W	
4	PINT4S	0	R/W	
3	PINT3S	0	R/W	
2	PINT2S	0	R/W	
1	PINT1S	0	R/W	
0	PINT0S	0	R/W	

[Legend]
n = 7 to 0

6.3.5 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 or in deep standby mode.

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

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

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

6.3.6 PINT Interrupt Enable Register (PINTER)

PINTER is a 16-bit register that enables interrupt request inputs to external interrupt input pins PINT7 to PINT0. PINTER is initialized by a power-on reset or in deep standby mode.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PINT7E	0	R/W	PINT Enable These bits select whether to enable interrupt request inputs to external interrupt input pins PINT7 to PINT0. 0: PINTn input interrupt request is disabled 1: PINTn input interrupt request is enabled
6	PINT6E	0	R/W	
5	PINT5E	0	R/W	
4	PINT4E	0	R/W	
3	PINT3E	0	R/W	
2	PINT2E	0	R/W	
1	PINT1E	0	R/W	
0	PINT0E	0	R/W	

[Legend]
n = 7 to 0

6.3.7 PINT Interrupt Request Register (PIRR)

PIRR is a 16-bit register that indicates interrupt requests from external input pins PINT7 to PINT0. PIRR is initialized by a power-on reset or in deep standby mode.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	PINT7R	0	R	PINT Interrupt Request These bits indicate the status of the PINT7 to PINT0 interrupt requests. 0: No interrupt request at PINTn pin 1: Interrupt request at PINTn pin
6	PINT6R	0	R	
5	PINT5R	0	R	
4	PINT4R	0	R	
3	PINT3R	0	R	
2	PINT2R	0	R	
1	PINT1R	0	R	
0	PINT0R	0	R	

[Legend]
n = 7 to 0

6.3.8 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 or in deep standby mode.

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

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

6.3.9 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 or in deep standby mode.

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.

Note: * Bits BN[3:0] are initialized at a manual reset.

6.3.10 DMA Transfer Request Enable Register 0 (DREQER0)

DMA transfer request enable register 0 (DREQER0) is an 8-bit readable/writable register that enables/disables the IIC3 DMA transfer requests, and enables/disables CPU interrupt requests.

DMA transfer request enable register 0 is initialized by a power-on reset or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	Reserved		IIC3 2ch TX	IIC3 2ch RX	IIC3 1ch TX	IIC3 1ch RX	IIC3 0ch TX	IIC3 0ch RX
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	Reserved	0	R/W	DMA Transfer Request Enable Bits These bits enable/disable DMA transfer requests, and enable/disable CPU interrupt requests. 0: DMA transfer request disabled, CPU interrupt request enabled 1: DMA transfer request enabled, CPU interrupt request disabled
6	Reserved	0	R/W	
5	IIC3 2ch TX	0	R/W	
4	IIC3 2ch RX	0	R/W	
3	IIC3 1ch TX	0	R/W	
2	IIC3 1ch RX	0	R/W	
1	IIC3 0ch TX	0	R/W	
0	IIC3 0ch RX	0	R/W	

6.3.11 DMA Transfer Request Enable Register 1 (DREQER1)

DMA transfer request enable register 1 (DREQER1) is an 8-bit readable/writable register that enables/disables the SCIF (channels 0 to 3) DMA transfer requests, and enables/disables CPU interrupt requests.

DMA transfer request enable register 1 is initialized by a power-on reset or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	SCIF 3ch TX	SCIF 3ch RX	SCIF 2ch TX	SCIF 2ch RX	SCIF 1ch TX	SCIF 1ch RX	SCIF 0ch TX	SCIF 0ch RX
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	SCIF 3ch TX	0	R/W	DMA Transfer Request Enable Bits These bits enable/disable DMA transfer requests, and enable/disable CPU interrupt requests. 0: DMA transfer request disabled, CPU interrupt request enabled 1: DMA transfer request enabled, CPU interrupt request disabled
6	SCIF 3ch RX	0	R/W	
5	SCIF 2ch TX	0	R/W	
4	SCIF 2ch RX	0	R/W	
3	SCIF 1ch TX	0	R/W	
2	SCIF 1ch RX	0	R/W	
1	SCIF 0ch TX	0	R/W	
0	SCIF 0ch RX	0	R/W	

6.3.12 DMA Transfer Request Enable Register 2 (DREQER2)

DMA transfer request enable register 2 (DREQER2) is an 8-bit readable/writable register that enables/disables the SCIF (channels 4 to 7) DMA transfer requests, and enables/disables CPU interrupt requests.

DMA transfer request enable register 2 is initialized by a power-on reset or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	SCIF 7ch TX	SCIF 7ch RX	SCIF 6ch TX	SCIF 6ch RX	SCIF 5ch TX	SCIF 5ch RX	SCIF 4ch TX	SCIF 4ch RX
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	SCIF 7ch TX	0	R/W	DMA Transfer Request Enable Bits These bits enable/disable DMA transfer requests, and enable/disable CPU interrupt requests. 0: DMA transfer request disabled, CPU interrupt request enabled 1: DMA transfer request enabled, CPU interrupt request disabled
6	SCIF 7ch RX	0	R/W	
5	SCIF 6ch TX	0	R/W	
4	SCIF 6ch RX	0	R/W	
3	SCIF 5ch TX	0	R/W	
2	SCIF 5ch RX	0	R/W	
1	SCIF 4ch TX	0	R/W	
0	SCIF 4ch RX	0	R/W	

6.3.13 DMA Transfer Request Enable Register 3 (DREQER3)

DMA transfer request enable register 3 (DREQER3) is an 8-bit readable/writable register that enables/disables the ADC, MTU2 (channels 0 to 4), and RCAN-ET (channels 0 and 1) DMA transfer requests, and enables/disables CPU interrupt requests.

DMA transfer request enable register 3 is initialized by a power-on reset or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADC	MTU2 4ch	MTU2 3ch	MTU2 2ch	MTU2 1ch	MTU2 0ch	RCAN-ET 1ch	RCAN-ET 0ch
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	ADC	0	R/W	DMA Transfer Request Enable Bits These bits enable/disable DMA transfer requests, and enable/disable CPU interrupt requests. 0: DMA transfer request disabled, CPU interrupt request enabled 1: DMA transfer request enabled, CPU interrupt request disabled
6	MTU2 4ch	0	R/W	
5	MTU2 3ch	0	R/W	
4	MTU2 2ch	0	R/W	
3	MTU2 1ch	0	R/W	
2	MTU2 0ch	0	R/W	
1	RCAN-ET 1ch	0	R/W	
0	RCAN-ET 0ch	0	R/W	

6.4 Interrupt Sources

There are six types of interrupt sources: NMI, user break, H-UDI, IRQ, PINT, 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 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 exception handling sets the I3 to I0 bits in SR to level 15. For H-UDI interrupts, see section 26, User Debugging Interface (H-UDI).

6.4.4 IRQ Interrupts

IRQ interrupts are input from pins IRQ7 to IRQ0. As regard to the setting method of pins IRQ7 to IRQ0, see section 23, Pin Function Controller (PFC). 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.

When restoring from the service routine of IRQ interrupt exception handling, execute the RTE instruction after an interrupt request has been cleared in the IRQ interrupt request register (IRQRR).

6.4.5 PINT Interrupts

PINT interrupts are input from pins PINT7 to PINT0. As regard to the setting method of pins PINT7 to PINT0, see section 23, Pin Function Controller (PFC). Input of the interrupt requests is enabled by the PINT enable bits (PINT7E to PINT0E) in the PINT interrupt enable register (PINTER). For the PINT7 to PINT0 interrupts, low-level or high-level detection can be selected individually for each pin by the PINT sense select bits (PINT7S to PINT0S) in interrupt control register 2 (ICR2). A single priority level in a range from 0 to 15 can be set for all PINT7 to PINT0 interrupts by bits 15 to 12 in interrupt priority register 05 (IPR05).

When using low-level sensing for the PINT7 to PINT0 interrupts, an interrupt request signal is sent to the INTC while the PINT7 to PINT0 pins are low. An interrupt request signal is stopped being sent to the INTC when the PINT7 to PINT0 pins are driven high. The status of the interrupt requests can be checked by reading the PINT interrupt request bits (PINT7R to PINT0R) in the PINT interrupt request register (PIRR). The above description also applies to when using high-level sensing, except for the polarity being reversed. The PINT interrupt exception handling sets the I3 to I0 bits in SR to the priority level of the PINT interrupt.

When restoring from the service routine of PINT interrupt exception handling, execute the RTE instruction after an interrupt request has been cleared in the PINT interrupt request register (PIRR).

6.4.6 On-Chip Peripheral Module Interrupts

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

- A/D converter (ADC)
- Multi-function timer pulse unit 2 (MTU2)
- Realtime clock (RTC)
- Watchdog timer (WDT)
- I^2C bus interface 3 (IIC3)
- Direct memory access controller (DMAC)
- Serial communication interface with FIFO (SCIF)
- Controller area network (RCAN-ET)
- Serial sound interface (SSI)
- 8-bit timer (TMR)

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 16 (IPR05 to IPR16). 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 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, Calculating Exception Handling Vector Table Addresses, in section 5, Exception Handling.

The priorities of IRQ interrupts, PINT 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 16 (IPR01, IPR02, and IPR05 to IPR16). However, if two or more interrupts specified by the same IPR among IPR05 to IPR16 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, PINT 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		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
		Vector	Vector Table Address Offset				
NMI		11	H'0000002C to H'0000002F	16	—	—	High
User break		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)	—	Low

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
		Vector	Vector Table Address Offset				
PINT	PINT0	80	H'00000140 to H'00000143	0 to 15 (0)	IPR05 (15 to 12)	1	High
	PINT1	81	H'00000144 to H'00000147			2	
	PINT2	82	H'00000148 to H'0000014B			3	
	PINT3	83	H'0000014C to H'0000014F			4	
	PINT4	84	H'00000150 to H'00000153			5	
	PINT5	85	H'00000154 to H'00000157			6	
	PINT6	86	H'00000158 to H'0000015B			7	
	PINT7	87	H'0000015C to H'0000015F			8	
ADC	ADI	92	H'00000170 to H'00000173	0 to 15 (0)	IPR05 (7 to 4)	—	Low

Interrupt Source			Interrupt Vector: Vector	Interrupt Vector: Vector Table Address Offset	Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
MTU2	MTU0	TGI0A	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPR06 (11 to 8)	1	High
		TGI0B	109	H'000001B4 to H'000001B7			2	
		TGI0C	110	H'000001B8 to H'000001BB			3	
		TGI0D	111	H'000001BC to H'000001BF			4	
		TCI0V	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPR06 (7 to 4)	1	
		TCI0E	113	H'000001C4 to H'000001C7			2	
		TCI0F	114	H'000001C8 to H'000001CB			3	
	MTU1	TGI1A	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPR06 (3 to 0)	1	
		TGI1B	117	H'000001D4 to H'000001D7			2	
		TCI1V	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPR07 (15 to 12)	1	
		TCI1U	121	H'000001E4 to H'000001E7			2	
	MTU2	TGI2A	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPR07 (11 to 8)	1	
		TGI2B	125	H'000001F4 to H'000001F7			2	
		TCI2V	128	H'00000200 to H'00000203	0 to 15 (0)	IPR07 (7 to 4)	1	
		TCI2U	129	H'00000204 to H'00000207			2	Low

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
			Vector	Vector Table Address Offset				
MTU2	MTU3	TGI3A	132	H'00000210 to H'00000213	0 to 15 (0)	IPR07 (3 to 0)	1	High
		TGI3B	133	H'00000214 to H'00000217			2	
		TGI3C	134	H'00000218 to H'0000021B			3	
		TGI3D	135	H'0000021C to H'0000021F			4	
		TCI3V	136	H'00000220 to H'00000223	0 to 15 (0)	IPR08 (15 to 12)	—	
	MTU4	TGI4A	140	H'00000230 to H'00000233	0 to 15 (0)	IPR08 (11 to 8)	1	
		TGI4B	141	H'00000234 to H'00000237			2	
		TGI4C	142	H'00000238 to H'0000023B			3	
		TGI4D	143	H'0000023C to H'0000023F			4	
		TCI4V	144	H'00000240 to H'00000243	0 to 15 (0)	IPR08 (7 to 4)	—	
	MTU5	TGI5U	148	H'00000250 to H'00000253	0 to 15 (0)	IPR08 (3 to 0)	1	
		TGI5V	149	H'00000254 to H'00000257			2	
		TGI5W	150	H'00000258 to H'0000025B			3	
RTC	ARM		152	H'00000260 to H'00000263	0 to 15 (0)	IPR09 (15 to 12)	1	
	PRD		153	H'00000264 to H'00000267			2	
	CUP		154	H'00000268 to H'0000026B			3	Low

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
			Vector	Vector Table Address Offset				
WDT	ITI		156	H'00000270 to H'00000273	0 to 15 (0)	IPR09 (11 to 8)	—	High
IIC3	IIC0	STPI0	157	H'00000274 to H'00000277	0 to 15 (0)	IPR09 (7 to 4)	1	
		NAKI0	158	H'00000278 to H'0000027B			2	
		RXI0	159	H'0000027C to H'0000027F			3	
		TXI0	160	H'00000280 to H'00000283			4	
		TEI0	161	H'00000284 to H'00000287			5	
	IIC1	STPI1	164	H'00000290 to H'00000293	0 to 15 (0)	IPR10 (15 to 12)	1	
		NAKI1	165	H'00000294 to H'00000297			2	
		RXI1	166	H'00000298 to H'0000029B			3	
		TXI1	167	H'0000029C to H'0000029F			4	
		TEI1	168	H'000002A0 to H'000002A3			5	
	IIC2	STPI2	170	H'000002A8 to H'000002AB	0 to 15 (0)	IPR10 (11 to 8)	1	
		NAKI2	171	H'000002AC to H'000002AF			2	
		RXI2	172	H'000002B0 to H'000002B3			3	
		TXI2	173	H'000002B4 to H'000002B7			4	
		TEI2	174	H'000002B8 to H'000002BB			5	Low

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
			Vector	Vector Table Address Offset				
DMAC	DMAC0	DMINT0	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPR10 (7 to 4)	—	High
	DMAC1	DMINT1	177	H'000002C4 to H'000002C7	0 to 15 (0)	IPR10 (3 to 0)	—	
	DMAC2	DMINT2	178	H'000002C8 to H'000002CB	0 to 15 (0)	IPR11 (15 to 12)	—	
	DMAC3	DMINT3	179	H'000002CC to H'000002CF	0 to 15 (0)	IPR11 (11 to 8)	—	
SCIF	SCIF0	BRI0	180	H'000002D0 to H'000002D3	0 to 15 (0)	IPR11 (7 to 4)	1	
		ERI0	181	H'000002D4 to H'000002D7			2	
		RXI0	182	H'000002D8 to H'000002DB			3	
		TXI0	183	H'000002DC to H'000002DF			4	
	SCIF1	BRI1	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPR11 (3 to 0)	1	
		ERI1	185	H'000002E4 to H'000002E7			2	
		RXI1	186	H'000002E8 to H'000002EB			3	
		TXI1	187	H'000002EC to H'000002EF			4	
	SCIF2	BRI2	188	H'000002F0 to H'000002F3	0 to 15 (0)	IPR12 (15 to 12)	1	
		ERI2	189	H'000002F4 to H'000002F7			2	
		RXI2	190	H'000002F8 to H'000002FB			3	
		TXI2	191	H'000002FC to H'000002FF			4	Low

Interrupt Source			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	192	H'00000300 to H'00000303	0 to 15 (0)	IPR12 (11 to 8)	1	High
		ERI3	193	H'00000304 to H'00000307			2	
		RXI3	194	H'00000308 to H'0000030B			3	
		TXI3	195	H'0000030C to H'0000030F			4	
	SCIF4	BRI4	196	H'00000310 to H'00000313	0 to 15 (0)	IPR12 (7 to 4)	1	
		ERI4	197	H'00000314 to H'00000317			2	
		RXI4	198	H'00000318 to H'0000031B			3	
		TXI4	199	H'0000031C to H'0000031F			4	
	SCIF5	BRI5	200	H'00000320 to H'00000323	0 to 15 (0)	IPR12 (3 to 0)	1	
		ERI5	201	H'00000324 to H'00000327			2	
		RXI5	202	H'00000328 to H'0000032B			3	
		TXI5	203	H'0000032C to H'0000032F			4	
	SCIF6	BRI6	204	H'00000330 to H'00000333	0 to 15 (0)	IPR13 (15 to 12)	1	
		ERI6	205	H'00000334 to H'00000337			2	
		RXI6	206	H'00000338 to H'0000033B			3	
		TXI6	207	H'0000033C to H'0000033F			4	Low

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
			Vector	Vector Table Address Offset				
SCIF	SCIF7	BRI7	208	H'00000340 to H'00000343	0 to 15 (0)	IPR13 (11 to 8)	1	High
		ERI7	209	H'00000344 to H'00000347			2	
		RXI7	210	H'00000348 to H'0000034B			3	
		TXI7	211	H'0000034C to H'0000034F			4	
DMAC	DMINTA		212	H'00000350 to H'00000353	0 to 15 (0)	IPR13 (7 to 4)	—	
	DMAC4	DMINT4	216	H'00000360 to H'00000363	0 to 15 (0)	IPR13 (3 to 0)	—	
	DMAC5	DMINT5	217	H'00000364 to H'00000367	0 to 15 (0)	IPR14 (15 to 12)	—	
	DMAC6	DMINT6	218	H'00000368 to H'0000036B	0 to 15 (0)	IPR14 (11 to 8)	—	
	DMAC7	DMINT7	219	H'0000036C to H'0000036F	0 to 15 (0)	IPR14 (7 to 4)	—	
RCAN-ET	RCAN-ET0	ERS	228	H'00000390 to H'00000393	0 to 15 (0)	IPR15 (11 to 8)	1	
		OVR	229	H'00000394 to H'00000397			2	
		SLE	230	H'00000398 to H'0000039B			3	
		RM0	231	H'0000039C to H'0000039F			4	
		RM1	232	H'000003A0 to H'000003A3			5	Low

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bit)	IPR Setting Unit Internal Priority	Default Priority
			Vector	Vector Table Address Offset				
RCAN-ET	RCAN-ET1	ERS	234	H'000003A8 to H'000003AB	0 to 15 (0)	IPR15 (7 to 4)	1	High
		OVR	235	H'000003AC to H'000003AF			2	
		SLE	236	H'000003B0 to H'000003B3			3	
		RM0	237	H'000003B4 to H'000003B7			4	
		RM1	238	H'000003B8 to H'000003BB			5	
SSI	SSI0		244	H'000003D0 to H'000003D3	0 to 15 (0)	IPR16 (15 to 12)	—	
	SSI1		245	H'000003D4 to H'000003D7	0 to 15 (0)	IPR16 (11 to 8)	—	
TMR	TMR0	CMIA0	246	H'000003D8 to H'000003DB	0 to 15 (0)	IPR16 (7 to 4)	1	
		CMIB0	247	H'000003DC to H'000003DF			2	
		OVI0	248	H'000003E0 to H'000003E3			3	
	TMR1	CMIA1	252	H'000003F0 to H'000003F3	0 to 15 (0)	IPR16 (3 to 0)	1	
		CMIB1	253	H'000003F4 to H'000003F7			2	
		OVI1	254	H'000003F8 to H'000003FB			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 16 (IPR01, IPR02, and IPR05 to IPR16). 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. 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).
5. The start address of the interrupt exception service routine is fetched from the exception handling vector table corresponding to the accepted interrupt.
6. 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.
7. The program counter (PC) is saved onto the stack.
8. The CPU jumps to the fetched start address of the interrupt exception service routine and starts executing the program. The jump that occurs is not a delayed branch.

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 or in deep standby mode.

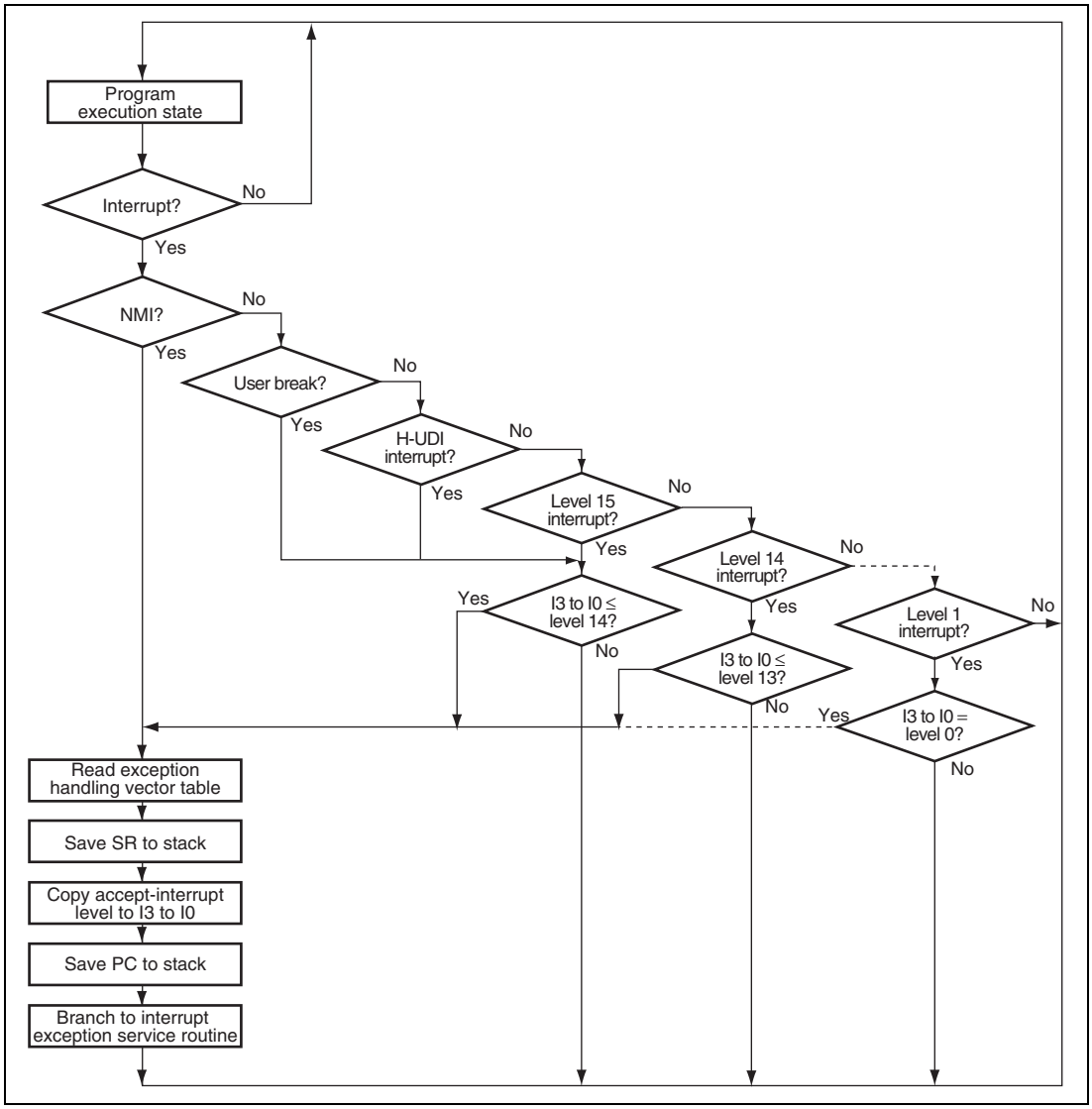


Figure 6.2 Interrupt Operation Flow

6.6.2 Stack after Interrupt Exception Handling

Figure 6.3 shows the stack after interrupt exception handling.

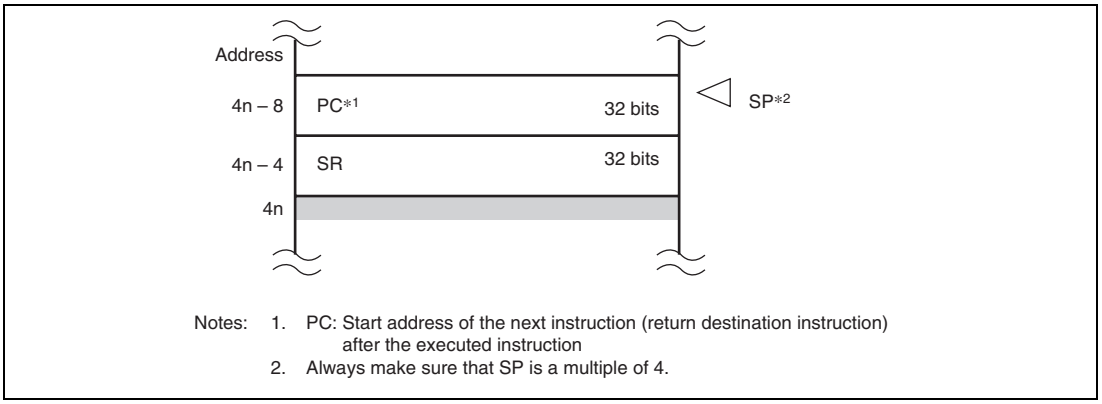


Figure 6.3 Stack after Interrupt Exception Handling

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 interrupt 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, PINT**	**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, PINT	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	120-MHz operation*1*2: 0.067 to 0.142 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	120-MHz operation*1*2: 0.100 to 0.175 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	120-MHz operation*1*2: 0.092 to 0.142 µs
		Max.	—	—	14 Icyc + 1 Pcyc + m1 + m2	14 Icyc + 3 Bcyc + 1 Pcyc + m1 + m2	14 Icyc + 1 Bcyc + 1 Pcyc + m1 + m2	120-MHz operation*1*2: 0.167 to 0.217 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	120-MHz operation*1*2: 0.092 to 0.142 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)	120-MHz operation*1*2: 0.245 to 0.300 µs

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

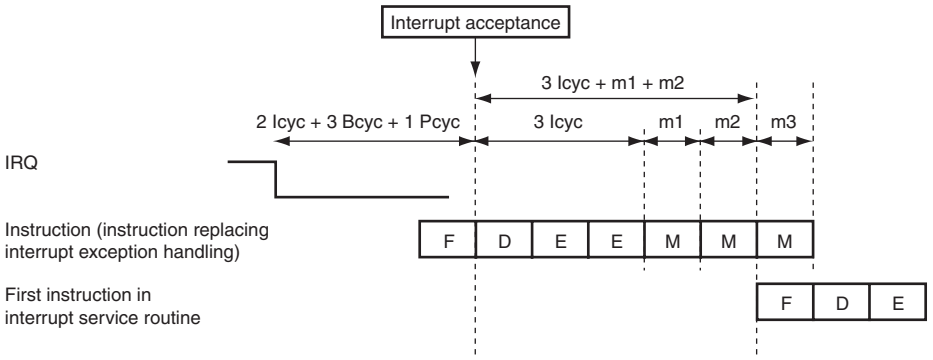
m1: Vector address read (longword read)

m2: SR save (longword write)

m3: PC save (longword write)

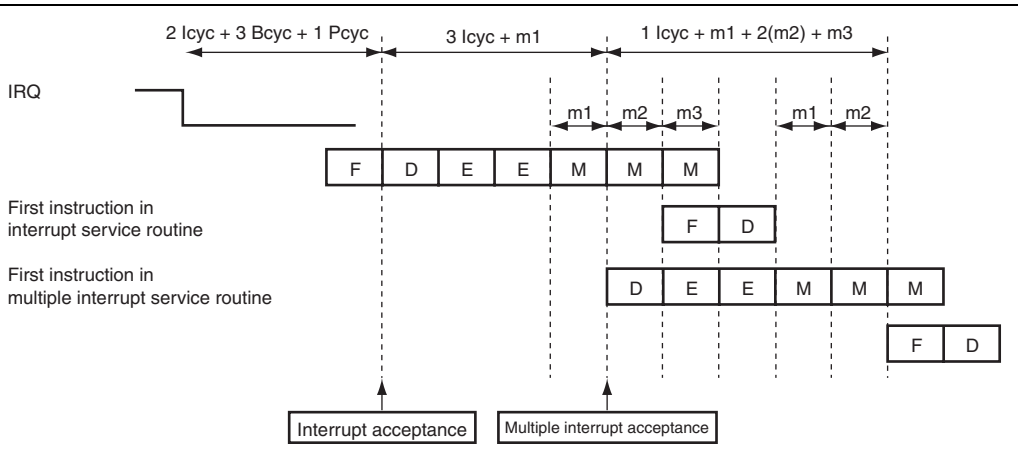
m4: Banked registers (R0 to R14, GBR, MACH, MACL, and PR) are restored from the stack.

1. In the case of m1 = m2 = m3 = m4 = 1 Icyc.
2. In the case of Iϕ:Bϕ:Pϕ = 120:60:30 [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)



[Legend]
m1: Vector address read
m2: Saving of SR (stack)
m3: Saving of PC (stack)

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

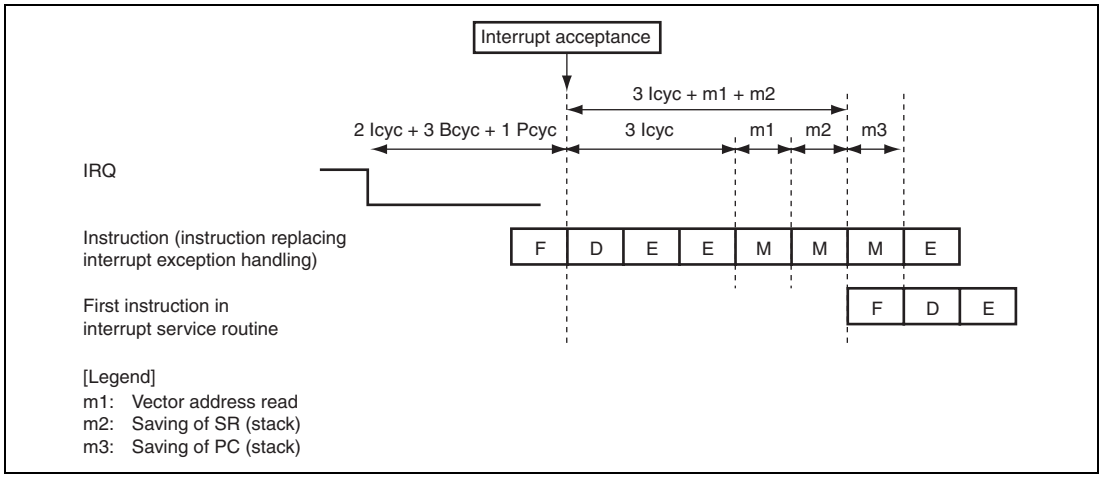


Figure 6.6 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking without Register Bank Overflow)

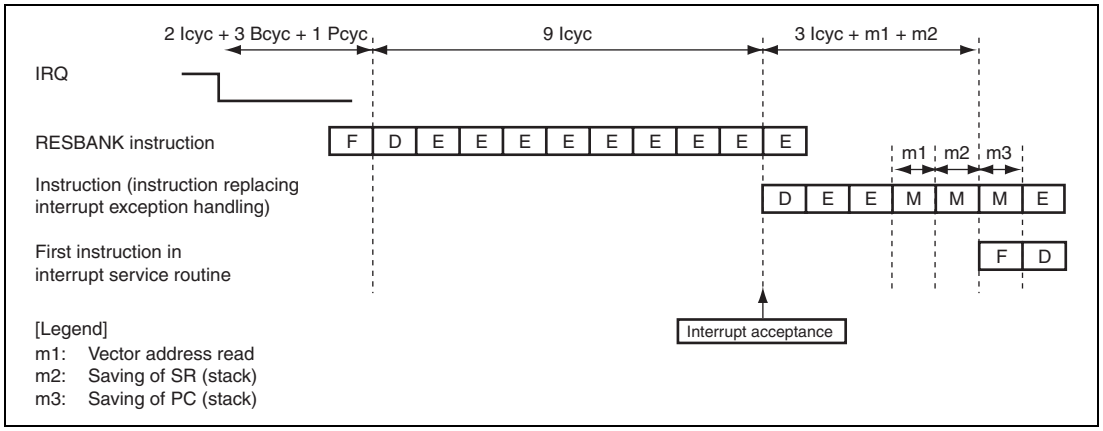


Figure 6.7 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking without Register Bank Overflow)

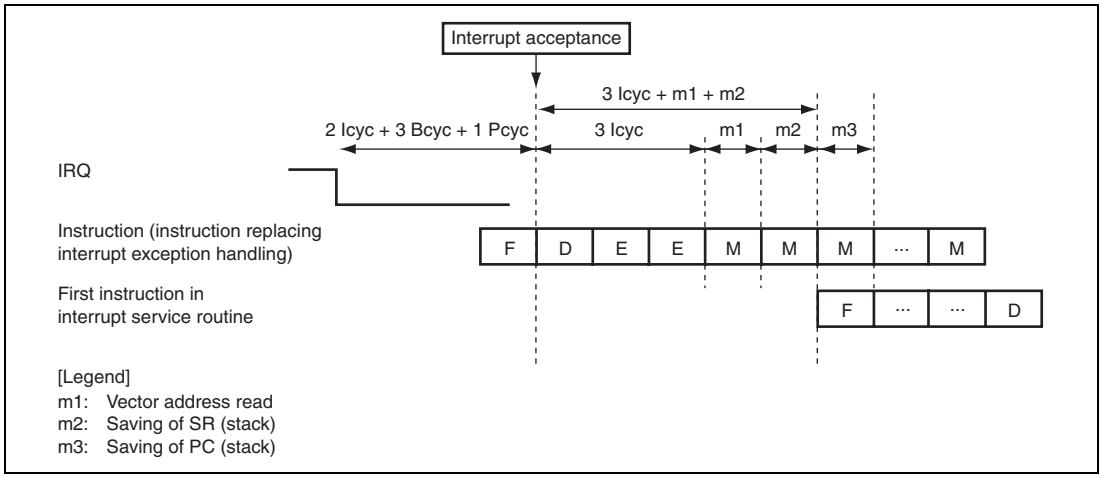


Figure 6.8 Example of Pipeline Operation when IRQ Interrupt is Accepted (Register Banking with Register Bank Overflow)

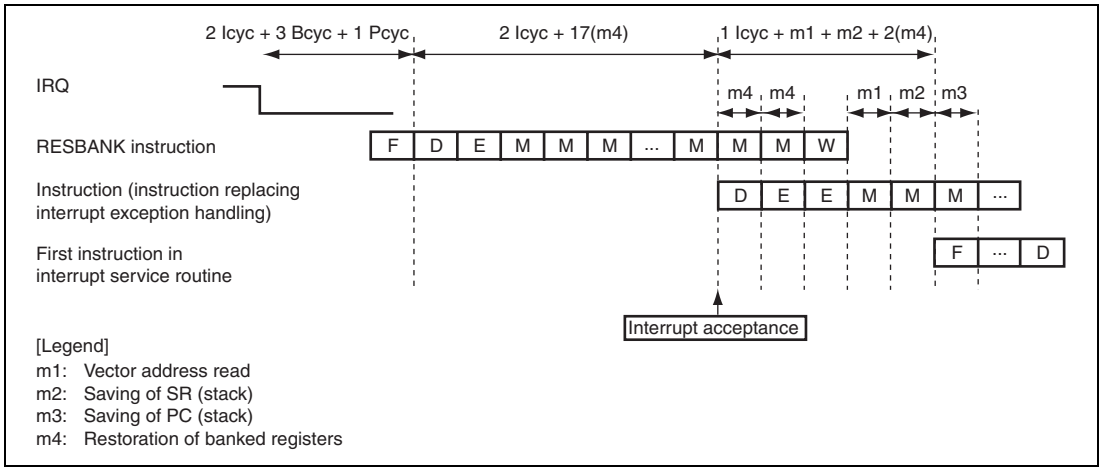


Figure 6.9 Example of Pipeline Operation when Interrupt is Accepted during RESBANK Instruction Execution (Register Banking with Register Bank Overflow)

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.

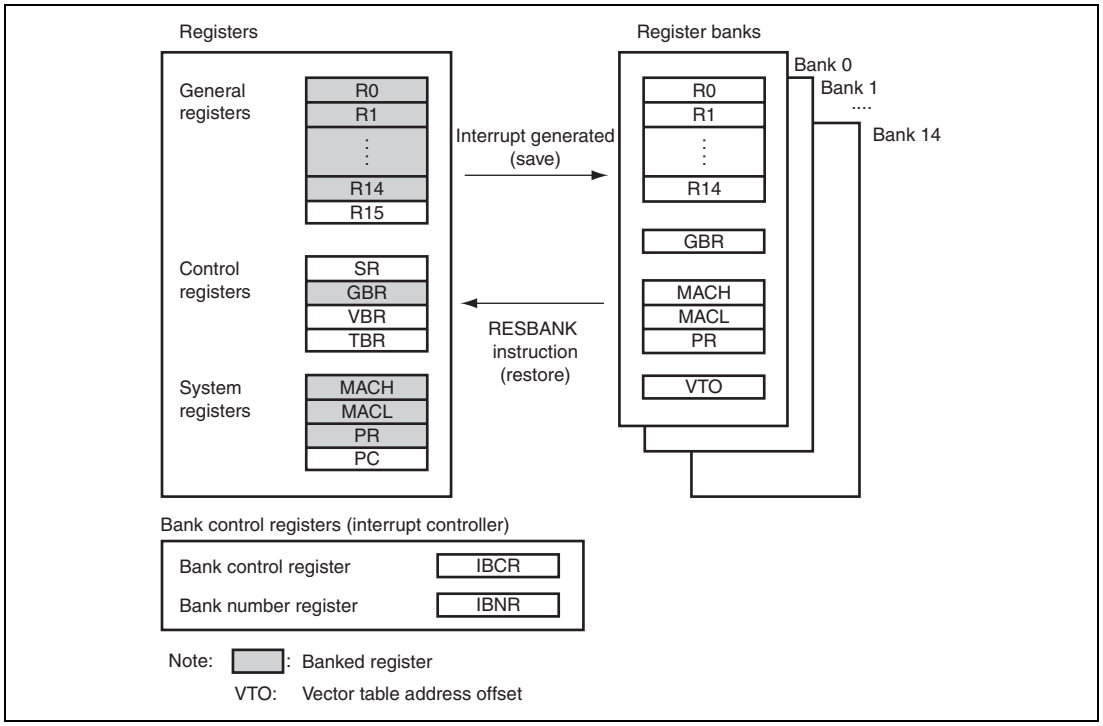


Figure 6.10 Overview of Register Bank Configuration

6.8.1 Register Banks and Bank Control Registers

(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 (VTO) are banked.

(2) Input/Output of 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.

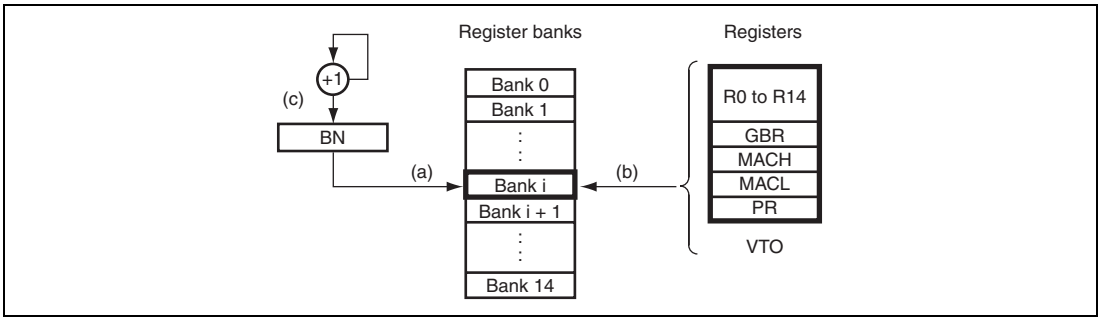


Figure 6.11 Bank Save Operations

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 exception service routine.

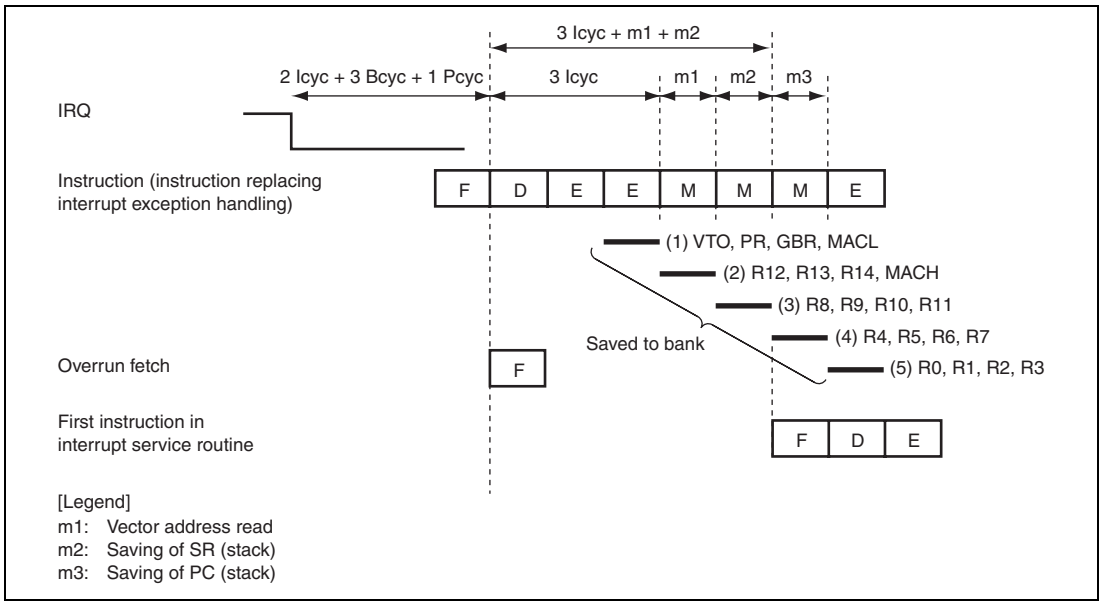


Figure 6.12 Bank Save Timing

(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 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 specified to activate the DMAC are masked by setting the DMA transfer enable bit in DREQER0 to DREQER3 to 1 without being input to the INTC.

6.9.1 Handling Interrupt Request Signals as Sources for CPU Interrupt but not DMAC Activation

1. Clear the corresponding DMAC transfer request enable bit in DREQER0 to DREQER3 to 0.
2. When an interrupt occurs, the interrupt request will be sent to the CPU.
3. The CPU clears the interrupt source and performs the necessary processing in the interrupt handling routine.

6.9.2 Handling Interrupt Request Signals as Sources for DMAC Activation but not CPU Interrupt

1. Select* the signals as DMAC activating sources by setting the corresponding DMAC transfer request enable bit in DREQER0 to DREQER3 to 1. This masks the CPU interrupt source regardless of the interrupt priority register settings.
2. When an interrupt occurs, the activation source will be sent to the DMAC.
3. The DMAC clears the activation source during the transfer.

Note: * As for the method to select the DMAC request sources, see section 11, Direct Memory Access Controller (DMAC).

6.10 Usage Note

6.10.1 Timing to Clear an Interrupt Source

The interrupt source flags 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.

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 of CPU, 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: two channels (channels 0 and 1)
   User break can be requested as the independent condition on channels 0 and 1.

- 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.
- Data
  Comparison of the 32-bit data is maskable in 1-bit units.
  One of the two data buses (M data bus (MDB) and I data bus (IDB)) can be selected.
- Bus cycle
  Instruction fetch (only when C bus is selected) or data access
- Read/write
- Operand size
  Byte, word, and longword

2. In an instruction fetch cycle, it can be selected whether the start of user break interrupt exception processing is set before or after an instruction is executed.
3. 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.

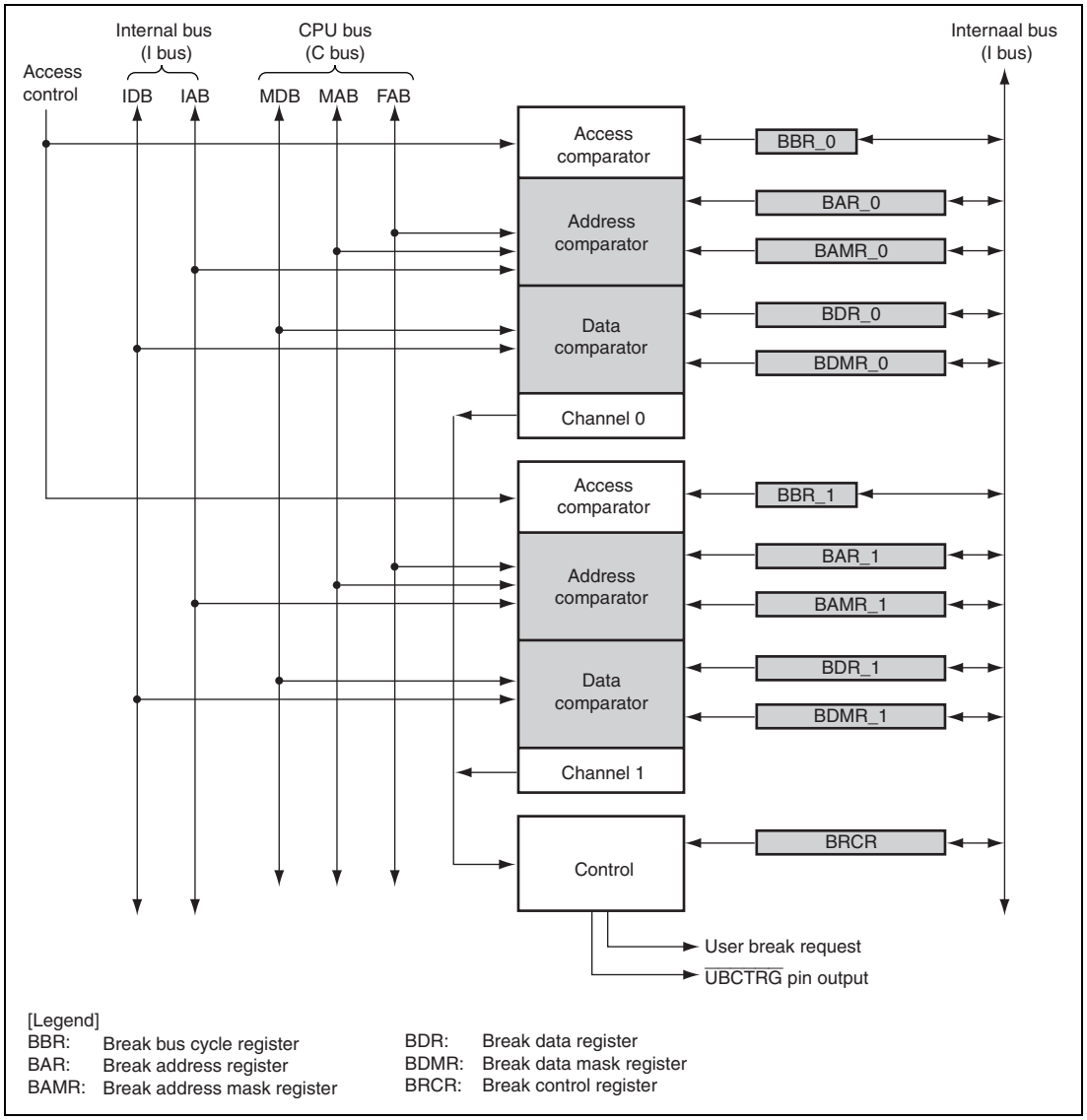


Figure 7.1 Block Diagram of UBC

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 or 1 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
	Break data register_0	BDR_0	R/W	H'00000000	H'FFFC0408	32
	Break data mask register_0	BDMR_0	R/W	H'00000000	H'FFFC040C	32
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
	Break data register_1	BDR_1	R/W	H'00000000	H'FFFC0418	32
	Break data mask register_1	BDMR_1	R/W	H'00000000	H'FFFC041C	32
Common	Break control register	BRCR	R/W	H'00000000	H'FFFC04C0	32

7.3.1 Break Address Register (BAR)

BAR is a 32-bit readable/writable register. BAR specifies the address used as a break condition in each channel. The control bits CD[1:0] in the break bus cycle register (BBR) select one of the three address buses for a break condition. BAR is initialized to H'00000000 by a power-on reset or in deep standby, 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
	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
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
	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
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	BA31 to BA0	All 0	R/W	Break Address Store an address on the CPU address bus (FAB or MAB) or IAB specifying break conditions. When the C bus and instruction fetch cycle are selected by BBR, specify an FAB address in bits BA31 to BA0. When the C bus and data access cycle are selected by BBR, specify an MAB address in bits BA31 to BA0.

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

7.3.2 Break Address Mask Register (BAMR)

BAMR is a 32-bit readable/writable register. BAMR specifies bits masked in the break address bits specified by BAR. BAMR is initialized to H'00000000 by a power-on reset or in deep standby, 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
	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24	BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
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
	BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8	BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
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	BAM31 to BAM0	All 0	R/W	Break Address Mask Specify bits masked in the break address bits specified by BAR (BA31 to BA0). 0: Break address bit BAn is included in the break condition 1: Break address bit BAn is masked and not included in the break condition Note: n = 31 to 0

7.3.3 Break Data Register (BDR)

BDR is a 32-bit readable/writable register. The control bits CD[1:0] in the break bus cycle register (BBR) select one of the two data buses for a break condition. BDR is initialized to H'00000000 by a power-on reset or in deep standby, 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
	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24	BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
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
	BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
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	BD31 to BD0	All 0	R/W	Break Data Bits Store data which specifies a break condition. If the I bus is selected in BBR, specify the break data on IDB in bits BD31 to BD0. If the C bus is selected in BBR, specify the break data on MDB in bits BD31 to BD0.

Notes: 1. Set the operand size when specifying a value on a data bus as the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 in BDR as the break data. Similarly, when the word size is selected, the same word data must be set in bits 31 to 16 and 15 to 0.

7.3.4 Break Data Mask Register (BDMR)

BDMR is a 32-bit readable/writable register. BDMR specifies bits masked in the break data bits specified by BDR. BDMR is initialized to H'00000000 by a power-on reset or in deep standby, 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
	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
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
	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8	BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
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	BDM31 to BDM0	All 0	R/W	Break Data Mask Specify bits masked in the break data bits specified by BDR (BD31 to BD0). 0: Break data bit BDn is included in the break condition 1: Break data bit BDn is masked and not included in the break condition Note: n = 31 to 0

Notes: 1. Set the operand size when specifying a value on a data bus as the break condition.
2. When the byte size is selected as a break condition, the same byte data must be set in bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 in BDMR as the break mask data. Similarly, when the word size is selected, the same word data must be set in bits 31 to 16 and 15 to 0.

7.3.5 Break Bus Cycle Register (BBR)

BBR 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. BBR is initialized to H'0000 by a power-on reset and in deep standby, 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
	—	—	UBID	DBE	CP[3:0]				CD[1:0]		ID[1:0]		RW[1:0]		SZ[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/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	UBID	0	R/W	User Break Interrupt Disable Disables or enables user break interrupt requests when a break condition is satisfied. 0: User break interrupt requests enabled 1: User break interrupt requests disabled
12	DBE	0	R/W	Data Break Enable Selects whether the data bus condition is included in the break conditions. 0: Data bus condition is not included in break conditions 1: Data bus condition is included in break conditions
11 to 8	CP[3:0]	0000	R/W	I-Bus Bus Select Select the bus master when the bus cycle of the break condition is the I bus cycle. However, when the C bus cycle is selected, this bit is invalidated (only the CPU cycle). xxx1: CPU cycle is included in break conditions xx1x: Reserved. Setting prohibited x1xx: Reserved. Setting prohibited 1xxx: Reserved. Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
7, 6	CD[1:0]	00	R/W	C Bus Cycle/I Bus Cycle Select Select the C bus cycle or I bus cycle as the bus cycle of the 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	ID[1:0]	00	R/W	Instruction Fetch/Data Access Select Select the instruction fetch cycle or data access cycle as the bus cycle of the 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	RW[1:0]	00	R/W	Read/Write Select Select the read cycle or write cycle as the bus cycle of the 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	SZ[1:0]	00	R/W	Operand Size Select Select the operand size of the bus cycle for the 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

7.3.6 Break Control Register (BRCR)

BRCR sets the following conditions:

1. Specifies whether a start of user break interrupt exception processing by instruction fetch cycle is 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 and in deep standby, 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	SCMFD 0	SCMFD 1	—	—	—	—	—	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	R	R	R	R	R/W	R/W	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	CKS[1:0]	00	R/W	Clock Select Specifies 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	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
12	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
11 to 7	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6	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
5	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
4 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 exception handling is described below:

1. The break address is set in the break address register (BAR). The masked address bits are set in the break address mask register (BAMR). The break data is set in the break data register (BDR). The masked data bits are set in the break data mask register (BDMR). 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 and the user break interrupt request is enabled, the UBC sends a user break request to the INTC, 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 CKS[1:0] 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. Clear the condition match flags during the user break interrupt exception processing routine. The interrupt occurs again if this operation is not performed.
5. There is a chance that the break set in channel 0 and the break set in channel 1 occur around the same time. In this case, there will be only one break request to the INTC, but these two break channel match flags may both be set.

6. When selecting the I bus as the break condition, note as follows:
   — Whether or not the access the CPU issued on the C bus is issued on the I bus depends on the setting of the cache. As regard to the I bus operation that depends on cache conditions, see table 8.8 in section 8, Cache.
   — When a break condition is specified for the I bus, only the data access cycle is monitored. The instruction fetch cycle (including cache update cycle) is not monitored.
   — 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 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 a start of user break interrupt exception processing is set before or after the execution of the instruction can be selected with the PCB0 or PCB1 bit in 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 user break interrupt request is not received 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 user break interrupt request is not received 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 logical addresses (and data) 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 addresses (and data) of the data access cycles on the bus specified by the I bus select bits, 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. When the data value is included in the break conditions:

   When the data value is included in the break conditions, either longword, word, or byte is specified as the operand size in the break bus cycle register (BBR). When data values are included in break conditions, a break is generated when the address conditions and data conditions both match. To specify byte data for this case, set the same data in the four bytes at bits 31 to 24, 23 to 16, 15 to 8, and 7 to 0 of the break data register (BDR) and break data mask register (BDMR). To specify word data for this case, set the same data in the two words at bits 31 to 16 and 15 to 0.
4. Access by a PREF instruction is handled as read access in longword units without access data. Therefore, if including the value of the data bus when a PREF instruction is specified as a break condition, a break will not occur.
5. 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 user break interrupt request is received, 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, BDR_1 = H'00000000, BDMR_1 = H'00000000, 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

  Data: H'00000000, Data mask: H'00000000

  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, BDR_1 = H'00000000, BDMR_1 = H'00000000, 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

  Data: H'00000000, Data 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

  BAR_0 = H'00008404, BAMR_0 = H'00000FFF, BBR_0 = H'0054, BAR_1= H'00008010, BAMR_1 = H'00000006, BBR_1 = H'0054, BDR_1 = H'00000000, BDMR_1 = H'00000000, 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

  Data: H'00000000, Data mask: H'00000000

  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

  BAR_0 = H'00123456, BAMR_0 = H'00000000, BBR_0 = H'0064, BAR_1= H'000ABCDE, BAMR_1 = H'000000FF, BBR_1 = H'106A, BDR_1 = H'A512A512, BDMR_1 = H'00000000, 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

  Data: H'0000A512, Data mask: H'00000000

  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 H'A512 is written in addresses H'000ABC00 to H'000ABCFE.

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

(Example 3-1)

- Register specifications

  BAR_0 = H'00314156, BAMR_0 = H'00000000, BBR_0 = H'0094, BAR_1= H'00055555, BAMR_1 = H'00000000, BBR_1 = H'11A9, BDR_1 = H'78787878, BDMR_1 = H'0F0F0F0F, 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

  Data: H'00000078, Data mask: H'0000000F

  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 CPU writes byte data H'7x in address H'00055555 on the I bus.

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 break after instruction execution for the DIVU or DIVS instruction. If a break after instruction execution is set for the DIVU or DIVS instruction and an exception or interrupt occurs during execution of the DIVU or DIVS instruction, a break after instruction execution occurs even though execution of the DIVU or DIVS instruction is halted.
10. Do not set a pre-execution break for the instruction that comes after the DIVU or DIVS instruction. If a pre-execution break is set for the instruction that comes after the DIVU or DIVS instruction and an exception or interrupt occurs during execution of the DIVU or DIVS instruction, a pre-execution break occurs even though execution of the DIVU or DIVS instruction is halted.
11. Do not set a pre-execution break and a break after instruction execution simultaneously in one address. For example, if a pre-execution break for channel 0 and a break after instruction execution for channel 1 are set simultaneously for one address, a break generated prior to instruction execution for channel 0 can set a condition-match flag after the instruction execution for channel 1.

Section 8 Cache

8.1 Features

- Capacity
  Instruction cache: 8 Kbytes
  Operand cache: 8 Kbytes
- Structure: Instructions/data separated, 4-way set associative
- Cache lock function (only for operand cache): Way 2 and way 3 are lockable
- Line size: 16 bytes
- Number of entries: 128 entries/way
- Write system: Write-back/write-through selectable
- Replacement method: Least-recently-used (LRU) algorithm

8.1.1 Cache Structure

The cache separates data and instructions and uses a 4-way set associative system. It is composed of four ways (banks), each of which is divided into an address section and a data section.

Each of the address and data sections is divided into 128 entries. The data section of the entry is called a line. Each line consists of 16 bytes (4 bytes × 4). The data capacity per way is 2 Kbytes (16 bytes × 128 entries), with a total of 8 Kbytes in the cache as a whole (4 ways). Figure 8.1 shows the operand cache structure. The instruction cache structure is the same as the operand cache structure except for not having the U bit.

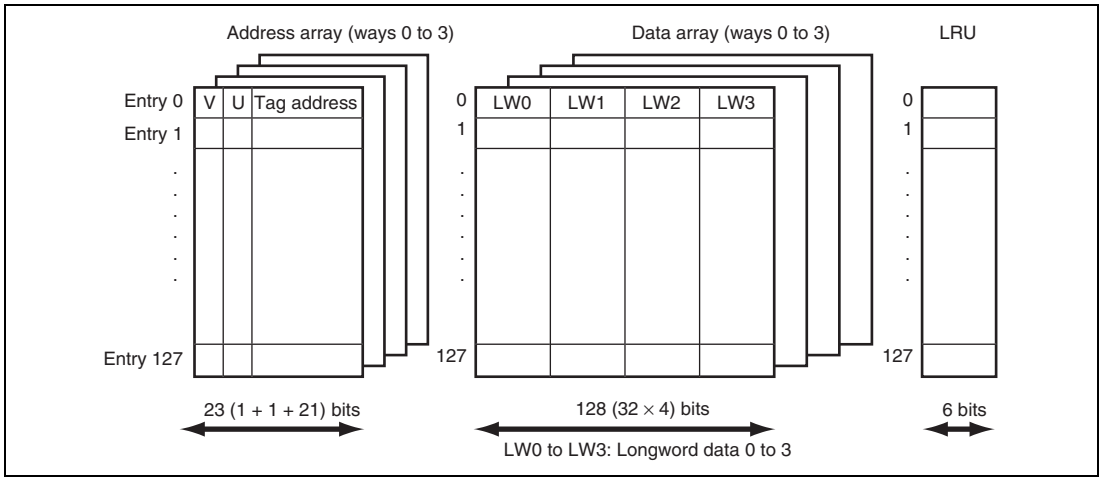


Figure 8.1 Operand Cache Structure

(1) Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, data is valid; when 0, data is not valid.

The U bit (only for operand cache) indicates whether the entry has been written to in write-back mode. When the U bit is 1, the entry has been written to; when 0, it has not.

The tag address holds the physical address used in the external memory access. It is composed of 21 bits (address bits 31 to 11) used for comparison during cache searches. In this LSI, as values of addresses in the cache valid space are from H'00000000 to H'1FFFFFFF (see section 9, Bus State Controller (BSC)), the upper three bits of the tag address are cleared to 0.

The V and U bits are initialized to 0 by a power-on reset and in deep standby mode but not initialized by a manual reset or in software standby mode.

The tag address is not initialized by a power-on reset or manual reset or in software standby mode. The tag address becomes undefined after deep standby.

(2) Data Array

Holds a 16-byte instruction or data. Entries are registered in the cache in line units (16 bytes).

The data array is not initialized by a power-on reset or manual reset or in software standby mode. The data array becomes undefined after deep standby.

(3) LRU

With the 4-way set associative system, up to four instructions or data with the same entry address can be registered in the cache. When an entry is registered, LRU shows which of the four ways it is recorded in. There are six LRU bits, controlled by hardware. A least-recently-used (LRU) algorithm is used to select the way that has been least recently accessed.

Six LRU bits indicate the way to be replaced in case of a cache miss. The relationship between LRU and way replacement is shown in table 8.1 when the cache lock function (only for operand cache) is not used (concerning the case where the cache lock function is used, see section 8.2.2, Cache Control Register 2 (CCR2)). If a bit pattern other than those listed in table 8.1 is set in the LRU bits by software, the cache will not function correctly. When modifying the LRU bits by software, set one of the patterns listed in table 8.1.

The LRU bits are initialized to B'000000 by a power-on reset and in deep standby but not initialized by a manual reset or in software standby mode.

Table 8.1 LRU and Way Replacement (Cache Lock Function Not Used)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000100, 010100, 100000, 110000, 110100	3
000001, 000011, 001011, 100001, 101001, 101011	2
000110, 000111, 001111, 010110, 011110, 011111	1
111000, 111001, 111011, 111100, 111110, 111111	0

8.2 Register Descriptions

The cache has the following registers.

Table 8.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Cache control register 1	CCR1	R/W	H'00000000	H'FFFC1000	32
Cache control register 2	CCR2	R/W	H'00000000	H'FFFC1004	32

8.2.1 Cache Control Register 1 (CCR1)

The instruction cache is enabled or disabled using the ICE bit. The ICF bit controls disabling of all instruction cache entries. The operand cache is enabled or disabled using the OCE bit. The OCF bit controls disabling of all operand cache entries. The WT bit selects either write-through mode or write-back mode for operand cache.

Programs that change the contents of CCR1 should be placed in an address space that is not cached, and an address space that is cached should be accessed after reading the contents of CCR1.

CCR1 is initialized to H'00000000 by a power-on reset and in deep standby but not initialized by a manual reset or 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
	—	—	—	—	ICF	—	—	ICE	—	—	—	—	OCF	—	WT	OCE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11	ICF	0	R/W	Instruction Cache Flush Writing 1 flushes all instruction cache entries (clears the V and LRU bits of all instruction cache entries to 0). Always reads 0. Write-back to external memory is not performed when the instruction cache is flushed.
10, 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ICE	0	R/W	Instruction Cache Enable Indicates whether the instruction cache function is enabled or disabled. 0: Instruction cache disabled 1: Instruction cache enabled
7 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3	OCF	0	R/W	Operand Cache Flush Writing 1 flushes all operand cache entries (clears the V, U, and LRU bits of all operand cache entries to 0). Always reads 0. Write-back to external memory is not performed when the operand cache is flushed.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1	WT	0	R/W	Write Through Selects write-back mode or write-through mode. 0: Write-back mode 1: Write-through mode
0	OCE	0	R/W	Operand Cache Enable Indicates whether the operand cache function is enabled or disabled. 0: Operand cache disabled 1: Operand cache enabled

8.2.2 Cache Control Register 2 (CCR2)

CCR2 is used to enable or disable the cache locking function for operand cache and is valid in cache locking mode only. In cache locking mode, the lock enable bit (the LE bit) in CCR2 is set to 1. In non-cache-locking mode, the cache locking function is invalid.

When a cache miss occurs in cache locking mode by executing the prefetch instruction (PREF @Rn), the line of data pointed to by Rn is loaded into the cache according to bits 9 and 8 (the W3LOAD and W3LOCK bits) and bits 1 and 0 (the W2LOAD and W2LOCK bits) in CCR2. The relationship between the setting of each bit and a way, to be replaced when the prefetch instruction is executed, are listed in table 8.3. On the other hand, when the prefetch instruction is executed and a cache hit occurs, new data is not fetched and the entry which is already enabled is held. For example, when the prefetch instruction is executed with W3LOAD = 1 and W3LOCK = 1 specified in cache locking mode while one-line data already exists in way 0 which is specified by Rn, a cache hit occurs and data is not fetched to way 3.

In the cache access other than the prefetch instruction in cache locking mode, ways to be replaced by bits W3LOCK and W2LOCK are restricted. The relationship between the setting of each bit in CCR2 and ways to be replaced are listed in table 8.4.

Programs that change the contents of CCR2 should be placed in an address space that is not cached, and an address space that is cached should be accessed after reading the contents of CCR2.

CCR2 is initialized to H'00000000 by a power-on reset and in deep standby but not initialized by a manual reset or in software standby mode.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	W3 LOAD*	W3 LOCK	—	—	—	—	—	—	W2 LOAD*	W2 LOCK
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	R	R	R	R	R	R/W	R/W

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	LE	0	R/W	Lock Enable Enables or disables the cache locking function. 0: Non-cache locking mode 1: Cache locking mode
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	W3LOAD*	0	R/W	Way 3 Load
8	W3LOCK	0	R/W	Way 3 Lock When a cache miss occurs by the prefetch instruction while W3LOAD = 1 and W3LOCK = 1 in cache locking mode, the data is always loaded into way 3. Under any other condition, the cache miss data is loaded into the way to which LRU points.
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	W2LOAD*	0	R/W	Way 2 Load
0	W2LOCK	0	R/W	Way 2 Lock When a cache miss occurs by the prefetch instruction while W2LOAD = 1 and W2LOCK =1 in cache locking mode, the data is always loaded into way 2. Under any other condition, the cache miss data is loaded into the way to which LRU points.

Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.3 Way to be Replaced when a Cache Miss Occurs in PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to be Replaced
0	x	x	x	x	Decided by LRU (table 8.1)
1	x	0	x	0	Decided by LRU (table 8.1)
1	x	0	0	1	Decided by LRU (table 8.5)
1	0	1	x	0	Decided by LRU (table 8.6)
1	0	1	0	1	Decided by LRU (table 8.7)
1	0	x	1	1	Way 2
1	1	1	0	x	Way 3

[Legend]
x: Don't care
Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.4 Way to be Replaced when a Cache Miss Occurs in Other than PREF Instruction

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	Way to be Replaced
0	x	x	x	x	Decided by LRU (table 8.1)
1	x	0	x	0	Decided by LRU (table 8.1)
1	x	0	x	1	Decided by LRU (table 8.5)
1	x	1	x	0	Decided by LRU (table 8.6)
1	x	1	x	1	Decided by LRU (table 8.7)

[Legend]
x: Don't care
Note: * The W3LOAD and W2LOAD bits should not be set to 1 at the same time.

Table 8.5 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=0)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000100, 010100, 100000, 100001, 110000, 110100	3
000011, 000110, 000111, 001011, 001111, 010110, 011110, 011111	1
101001, 101011, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 8.6 LRU and Way Replacement (when W2LOCK=0 and W3LOCK=1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 001011, 100000, 100001, 101001, 101011	2
000100, 000110, 000111, 001111, 010100, 010110, 011110, 011111	1
110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

Table 8.7 LRU and Way Replacement (when W2LOCK=1 and W3LOCK=1)

LRU (Bits 5 to 0)	Way to be Replaced
000000, 000001, 000011, 000100, 000110, 000111, 001011, 001111, 010100, 010110, 011110, 011111	1
100000, 100001, 101001, 101011, 110000, 110100, 111000, 111001, 111011, 111100, 111110, 111111	0

8.3 Operation

Operations for the operand cache are described here. Operations for the instruction cache are similar to those for the operand cache except for the address array not having the U bit, and there being no prefetch operation or write operation, or a write-back buffer.

8.3.1 Searching Cache

If the operand cache is enabled (OCE bit in CCR1 is 1), whenever data in a cache-enabled area is accessed, the cache will be searched to see if the desired data is in the cache. Figure 8.2 illustrates the method by which the cache is searched.

Entries are selected using bits 10 to 4 of the address used to access memory and the tag address of that entry is read. At this time, the upper three bits of the tag address are always cleared to 0. Bits 31 to 11 of the address used to access memory are compared with the read tag address. The address comparison uses all four ways. When the comparison shows a match and the selected entry is valid (V = 1), a cache hit occurs. When the comparison does not show a match or the selected entry is not valid (V = 0), a cache miss occurs. Figure 8.2 shows a hit on way 1.

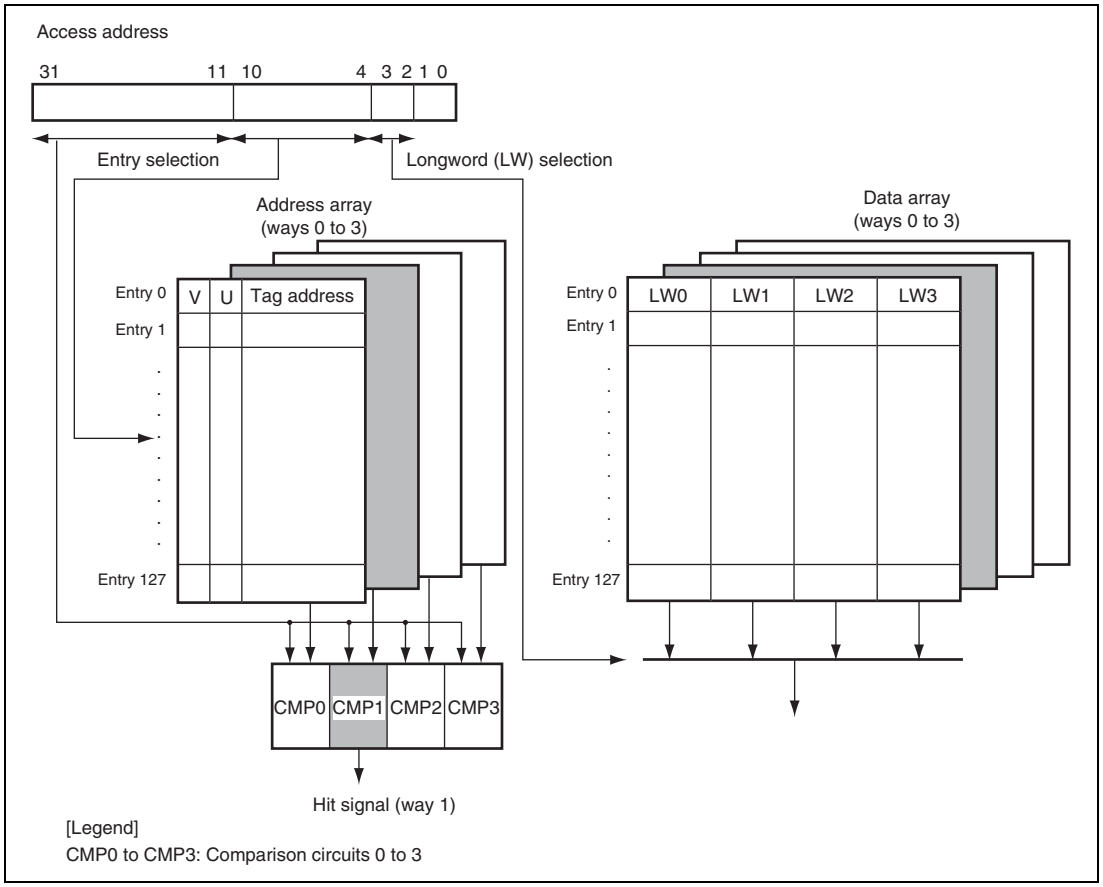


Figure 8.2 Cache Search Scheme

8.3.2 Read Access

(1) Read Hit

In a read access, data is transferred from the cache to the CPU. LRU is updated so that the hit way is the latest.

(2) Read Miss

An external bus cycle starts and the entry is updated. The way replaced follows table 8.4. Entries are updated in 16-byte units. When the desired data that caused the miss is loaded from external memory to the cache, the data is transferred to the CPU in parallel with being loaded to the cache. When it is loaded in the cache, the V bit is set to 1, and LRU is updated so that the replaced way becomes the latest. In operand cache, the U bit is additionally cleared to 0. When the U bit of the entry to be replaced by updating the entry in write-back mode is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes.

8.3.3 Prefetch Operation (Only for Operand Cache)

(1) Prefetch Hit

LRU is updated so that the hit way becomes the latest. The contents in other caches are not modified. No data is transferred to the CPU.

(2) Prefetch Miss

No data is transferred to the CPU. The way to be replaced follows table 8.3. Other operations are the same in case of read miss.

8.3.4 Write Operation (Only for Operand Cache)

(1) Write Hit

In a write access in write-back mode, the data is written to the cache and no external memory write cycle is issued. The U bit of the entry written is set to 1 and LRU is updated so that the hit way becomes the latest.

In write-through mode, the data is written to the cache and an external memory write cycle is issued. The U bit of the written entry is not updated and LRU is updated so that the replaced way becomes the latest.

(2) Write Miss

In write-back mode, an external bus cycle starts when a write miss occurs, and the entry is updated. The way to be replaced follows table 8.4. When the U bit of the entry to be replaced is 1, the cache update cycle starts after the entry is transferred to the write-back buffer. Data is written to the cache, the U bit is set to 1, and the V bit is set to 1. LRU is updated so that the replaced way becomes the latest. After the cache completes its update cycle, the write-back buffer writes the entry back to the memory. The write-back unit is 16 bytes.

In write-through mode, no write to cache occurs in a write miss; the write is only to the external memory.

8.3.5 Write-Back Buffer (Only for Operand Cache)

When the U bit of the entry to be replaced in the write-back mode is 1, it must be written back to the external memory. To increase performance, the entry to be replaced is first transferred to the write-back buffer and fetching of new entries to the cache takes priority over writing back to the external memory. After the cache completes to fetch the new entry, the write-back buffer writes the entry back to external memory. During the write-back cycles, the cache can be accessed. The write-back buffer can hold one line of cache data (16 bytes) and its physical address. Figure 8.3 shows the configuration of the write-back buffer.

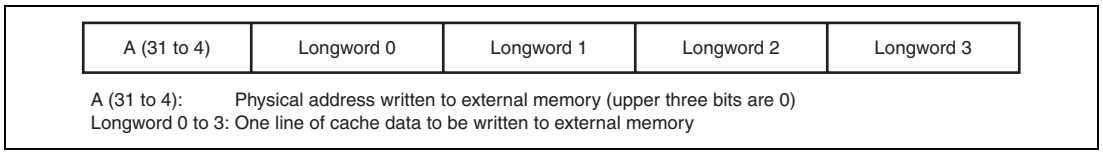


Figure 8.3 Write-Back Buffer Configuration

Operations in sections 8.3.2 to 8.3.5 are compiled in table 8.8

Table 8.8 Cache Operations

Cache	CPU Cycle	Hit/ miss	Write-back mode/ write through mode	U Bit	External Memory Accession (through internal bus)	Cache Contents
Instruction cache	Instruction fetch	Hit	—	—	Not generated	Not renewed
		Miss	—	—	Cache renewal cycle is generated.	Renewed to new values by cache renewal cycle
Operand cache	Prefetch/ read	Hit	Either mode is available	x	Not generated	Not renewed
		Miss	Write-through mode	—	Cache renewal cycle is generated.	Renewed to new values by cache renewal cycle
			Write-back mode	0	Cache renewal cycle is generated	Renewed to new values by cache renewal cycle
				1	Cache renewal cycle is generated. Succeedingly write-back cycle in write-back buffer is generated	Renewed to new values by cache renewal cycle
	Write	Hit	Write-through mode	—	Write cycle CPU issues is generated.	Renewed to new values by write cycle the CPU issues
			Write-back mode	x	Not generated	Renewed to new values by write cycle the CPU issues
		Miss	Write-through mode	—	Write cycle CPU issues is generated.	Not renewed*
			Write-back mode	0	Cache renewal cycle is generated.	Renewed to new values by cache renewal cycle. Subsequently renewed again to new values in write cycle CPU issues.
				1	Cache renewal cycle is generated. Succeedingly write-back cycle in write-back buffer is generated	Renewed to new values by cache renewal cycle. Subsequently renewed again to new values in write cycle CPU issues.

[Legend]

x: Don't care

Note: Cache renewal cycle: 16-byte read access, write-back cycle in write-back buffer: 16-byte write access

* Neither LRU renewed. LRU is renewed in all other cases.

8.3.6 Coherency of Cache and External Memory

Use software to ensure coherency between the cache and the external memory. When memory shared by this LSI and another device is mapped in the address space to be cached, operate the memory-mapped cache to invalidate and write back as required.

8.4 Memory-Mapped Cache

To allow software management of the cache, cache contents can be read and written by means of MOV instructions. The instruction cache address array is mapped onto addresses H'F0000000 to H'F07FFFFF, and the data array onto addresses H'F1000000 to H'F17FFFFF. The operand cache address array is mapped onto addresses H'F0800000 to H'F0FFFFFF, and the data array onto addresses H'F1800000 to H'F1FFFFFF. Only longword can be used as the access size for the address array and data array, and instruction fetches cannot be performed.

8.4.1 Address Array

To access an address array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified.

In the address field, specify the entry address selecting the entry, the W bit for selecting the way, and the A bit for specifying the existence of associative operation. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the address array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

The tag address, LRU bits, U bit (only for operand cache), and V bit are specified as data. Always specify 0 for the upper three bits (bits 31 to 29) of the tag address.

For the address and data formats, see figure 8.4.

The following three operations are possible for the address array.

(1) Address Array Read

The tag address, LRU bits, U bit (only for operand cache), and V bit are read from the entry address specified by the address and the entry corresponding to the way. For the read operation, associative operation is not performed regardless of whether the associative bit (A bit) specified by the address is 1 or 0.

(2) Address-Array Write (Non-Associative Operation)

When the associative bit (A bit) in the address field is cleared to 0, write the tag address, LRU bits, U bit (only for operand cache), and V bit, specified by the data field, to the entry address specified by the address and the entry corresponding to the way. When writing to a cache line for which the U bit = 1 and the V bit =1 in the operand cache address array, write the contents of the cache line back to memory, then write the tag address, LRU bits, U bit, and V bit specified by the data field. When 0 is written to the V bit, 0 must also be written to the U bit of that entry.

(3) Address-Array Write (Associative Operation)

When writing with the associative bit (A bit) of the address field set to 1, the addresses in the four ways for the entry specified by the address field are compared with the tag address that is specified by the data field. Write the U bit (only for operand cache) and the V bit specified by the data field to the entry of the way that has a hit. However, the tag address and LRU bits remain unchanged. When there is no way that has a hit, nothing is written and there is no operation.

This function is used to invalidate a specific entry in the cache. When the U bit of the entry that has had a hit is 1 in the operand cache, writing back should be performed. However, when 0 is written to the V bit, 0 must also be written to the U bit of that entry.

8.4.2 Data Array

To access a data array, the 32-bit address field (for read/write accesses) and 32-bit data field (for write accesses) must be specified. The address field specifies information for selecting the entry to be accessed; the data field specifies the longword data to be written to the data array.

Specify the entry address for selecting the entry, the L bit indicating the longword position within the (16-byte) line, and the W bit for selecting the way. In the L bit, B'00 is longword 0, B'01 is longword 1, B'10 is longword 2, and B'11 is longword 3. In the W bit, B'00 is way 0, B'01 is way 1, B'10 is way 2, and B'11 is way 3. Since the access size of the data array is fixed at longword, specify B'00 for bits 1 and 0 of the address.

For the address and data formats, see figure 8.4.

The following two operations are possible for the data array. Information in the address array is not modified by this operation.

(1) Data Array Read

The data specified by the L bit in the address is read from the entry address specified by the address and the entry corresponding to the way.

(2) Data Array Write

The longword data specified by the data is written to the position specified by the L bit in the address from the entry address specified by the address and the entry corresponding to the way.

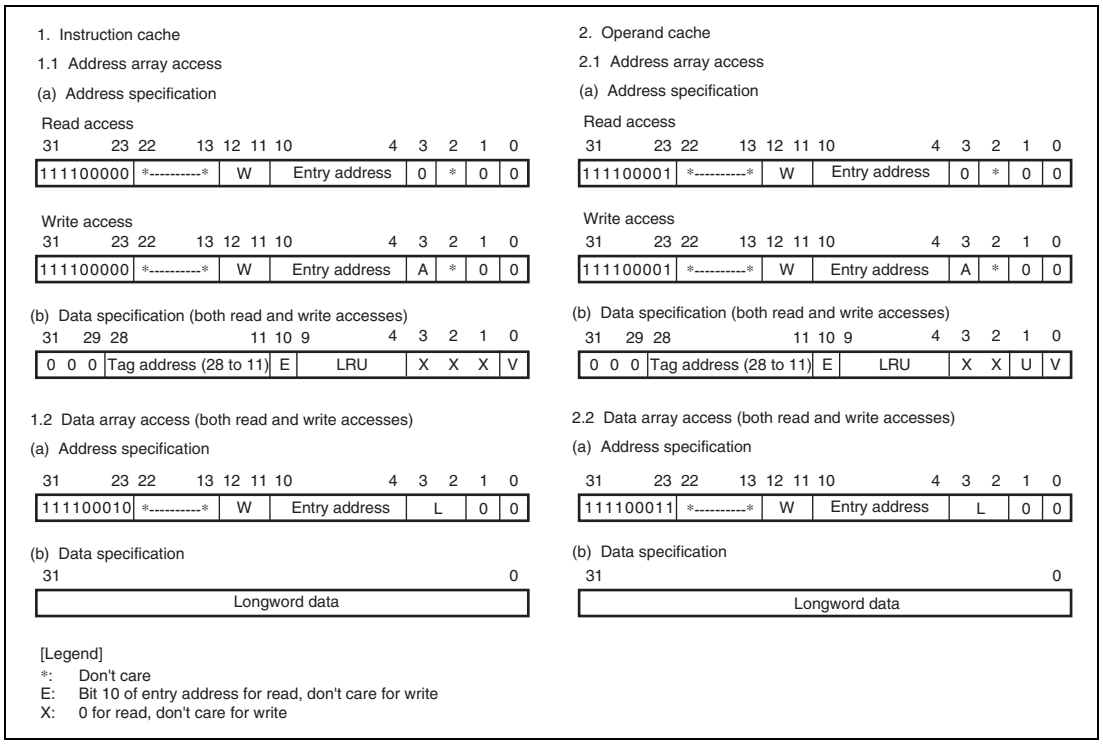


Figure 8.4 Specifying Address and Data for Memory-Mapped Cache Access

8.4.3 Usage Examples

(1) Invalidating Specific Entries

Specific cache entries can be invalidated by writing 0 to the entry's V bit in the memory mapping cache access. When the A bit is 1, the tag address specified by the write data is compared to the tag address within the cache selected by the entry address, and data is written to the bits V and U specified by the write data when a match is found. If no match is found, there is no operation. When the V bit of an entry in the address array is set to 0, the entry is written back if the entry's U bit is 1.

An example when a write data is specified in R0 and an address is specified in R1 is shown below.

```
; R0=H'0110 0010; tag address(28-11)=B'0 0001 0001 0000 0000 0, U=0, V=0
; R1=H'F080 0088; operand cache address array access, entry=B'000 1000, A=1
;
 MOV.L  R0,@R1
```

(2) Reading the Data of a Specific Entry

The data section of a specific cache entry can be read by the memory mapping cache access. The longword indicated in the data field of the data array in figure 8.4 is read into the register.

An example when an address is specified in R0 and data is read in R1 is shown below.

```
; R0=H'F100 004C; instruction cache data array access, entry=B'000 0100,
; Way=0, longword address=3
;
 MOV.L  @R0,R1
```

8.4.4 Notes

1. Programs that access memory-mapped cache of the operand cache should be placed in a cache-disabled space. Programs that access memory-mapped cache of the instruction cache should be placed in a cache-disabled space, and in each of the beginning and the end of that, two or more read accesses to on-chip peripheral modules or external address space (cache-disabled address) should be executed.
2. Rewriting the address array contents so that two or more ways are hit simultaneously is prohibited. Operation is not guaranteed if the address array contents are changed so that two or more ways are hit simultaneously.
3. Memory-mapped cache can be accessed only by the CPU and not by the DMAC. Registers can be accessed by the CPU and the DMAC.

Section 9 Bus State Controller (BSC)

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

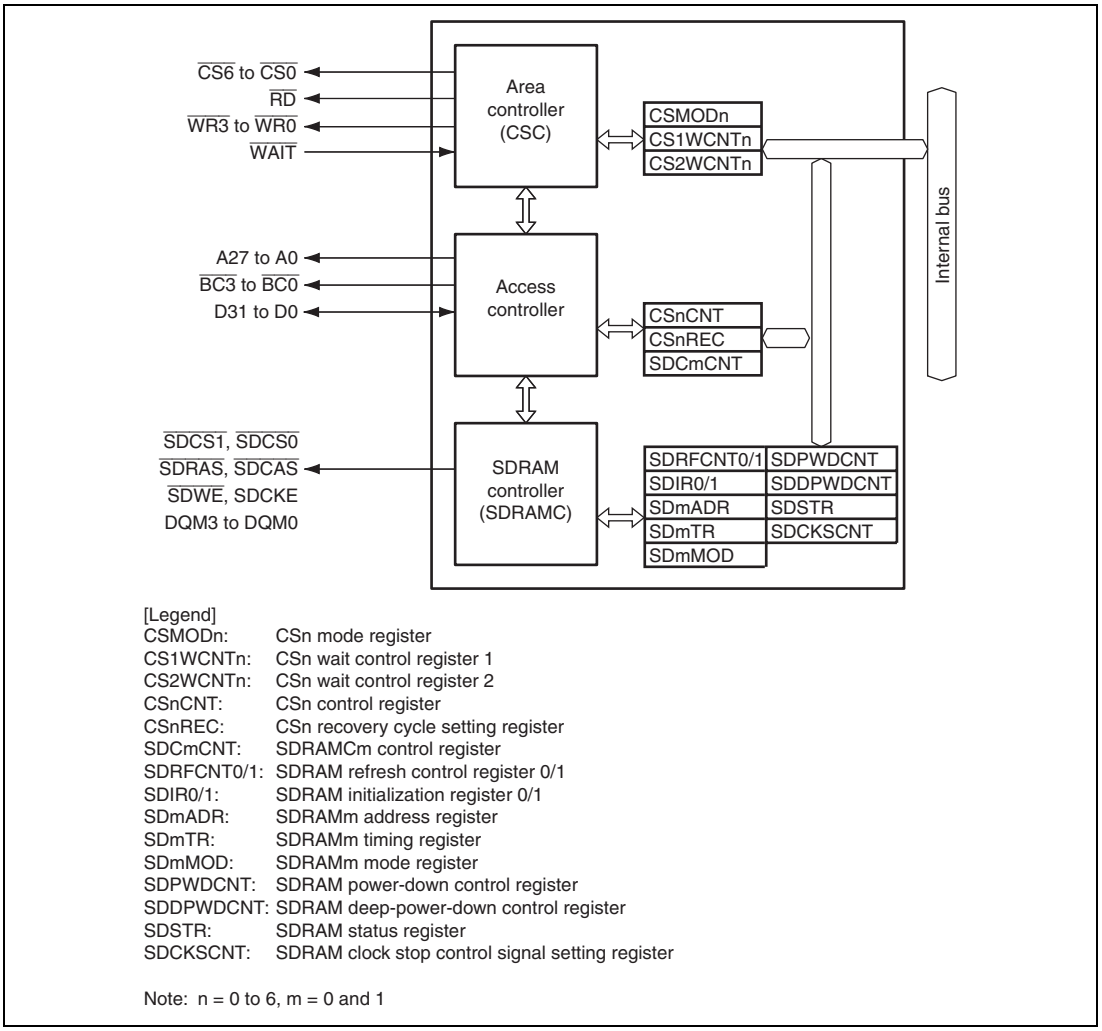
9.1 Features

1. External address space
- A maximum of 64 Mbytes for the SDRAM and each for areas CS0 to CS6 (256 Mbytes for CS6)
- Ability to select the data bus width (8, 16, or 32 bits) independently for each address space

2. Normal space interface
- Supports an interface for direct connection to SRAM
- Cycle wait function: Maximum of 31 wait states (maximum of seven wait states for page access cycles)
- Wait control
  - — Ability to select the assert/negate timing for chip select signals
  - — Ability to select the assert/negate timing for the read strobe and write strobe signals
  - — Ability to select the data output start/end timing
  - — Ability to select the delay for chip select signals
- Write access modes: One-write strobe and byte-write strobe modes
- Page access mode: Support for page read and page write (64-bit, 128-bit, and 256-bit page units)

3. SDRAM interface
- Ability to set SDRAM in up to two areas
- Refresh functions
  - — Auto-refresh (on-chip programmable refresh counter)
  - — Self-refresh
- Ability to select the access timing (support for low column latency, column latency, and low active interval settings)
- Initialization sequencer function, power-down function, deep-power-down function, and mode register setting function implemented on-chip

Figure 9.1 shows a block diagram of the BSC.



[Legend]

CSMODn:	CSn mode register
CS1WCNTn:	CSn wait control register 1
CS2WCNTn:	CSn wait control register 2
CSnCNT:	CSn control register
CSnREC:	CSn recovery cycle setting register
SDCmCNT:	SDRAMCm control register
SDRFCNT0/1:	SDRAM refresh control register 0/1
SDIR0/1:	SDRAM initialization register 0/1
SDmADR:	SDRAMm address register
SDmTR:	SDRAMm timing register
SDmMOD:	SDRAMm mode register
SDPWDCNT:	SDRAM power-down control register
SDDPWDCNT:	SDRAM deep-power-down control register
SDSTR:	SDRAM status register
SDCKSCNT:	SDRAM clock stop control signal setting register

Note: n = 0 to 6, m = 0 and 1

Figure 9.1 Block Diagram of BSC

9.2 Input/Output Pins

Table 9.1 shows the pin configuration of the BSC.

Table 9.1 Pin Configuration

Name	I/O	Function
A27 to A0	Output	Address bus
D31 to D0	I/O	Data bus
$\overline{CS6}$ to $\overline{CS0}$	Output	Chip select
$\overline{RD}$	Output	Read pulse signal (read data output enable signal)
$\overline{WR3}$	Output	When accessing the 32-bit bus area, indicates that D31 to D24 are being written to in byte-write mode.
$\overline{WR2}$	Output	When accessing the 32-bit bus area, indicates that D23 to D16 are being written to in byte-write mode.
$\overline{WR1}$	Output	When accessing the 32-bit bus area, indicates that D15 to D8 are being written to in byte-write mode. When accessing the 16-bit bus area, indicates that D15 to D8 are being written to in byte-write mode.
$\overline{WR0}$	Output	When accessing the 8-bit bus area, indicates that D7 to D0 are being written to in byte-write mode.
$\overline{BC3}$	Output	When accessing the 32-bit bus area, indicates that D31 to D24 are being accessed in byte-access mode.
$\overline{BC2}$	Output	When accessing the 32-bit bus area, indicates that D23 to D16 are being written to in byte-write mode.
$\overline{BC1}$	Output	When accessing the 32-bit bus area, indicates that D15 to D8 are being accessed in byte-access mode. When accessing the 16-bit bus area, indicates that D15 to D8 are being accessed in byte-access mode.
$\overline{BC0}$	Output	When accessing the 8-bit bus area, indicates that D7 to D0 are being accessed in byte-access mode.
$\overline{SDCS1}$, $\overline{SDCS0}$	Output	Connects to $\overline{CS}$ pin when SDRAM is connected.
$\overline{SDRAS}$	Output	Connects to $\overline{RAS}$ pin when SDRAM is connected.
$\overline{SDCAS}$	Output	Connects to $\overline{CAS}$ pin when SDRAM is connected.
$\overline{SDWE}$	Output	Connects to $\overline{WE}$ pin when SDRAM is connected.

Name	I/O	Function
SDCKE	Output	Connects to CKE pin when SDRAM is connected.
DQM3	Output	Connects to DQMUU pin when SDRAM is connected by 32-bit SDRAM.
DQM2	Output	Connects to DQMUL pin when SDRAM is connected by 32-bit SDRAM.
DQM1	Output	Connects to DQMLU pin when SDRAM is connected by 32-bit bus. Connects to DQMU pin when SDRAM is connected by 16-bit bus.
DQM0	Output	Connects to DQMLL pin when SDRAM is connected by 32-bit bus. Connects to DQML pin when SDRAM is connected by 16-bit bus. Connects to DQM pin when SDRAM is connected by 8-bit bus.
$\overline{\text{WAIT}}$	Input	External wait input

9.3 Area Overview

9.3.1 Address Map

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

External address spaces CS5 to CS0 are cache-enabled when internal address A29 = 0 and cache-disabled when A29 = 1. The CS6 space is always cache-disabled.

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

Table 9.2 Address Map

Internal Address	Space	Memory to be Connected	Cache
H'00000000 to H'03FFFFFF	CS0	Normal space	Cache-enabled
H'04000000 to H'07FFFFFF	CS1	Normal space	
H'08000000 to H'0BFFFFFF	SDRAM0	SDRAM	
H'0C000000 to H'0FFFFFFF	SDRAM1	SDRAM	
H'10000000 to H'13FFFFFF	CS2	Normal space	
H'14000000 to H'17FFFFFF	CS3	Normal space	
H'18000000 to H'1BFFFFFF	CS4	Normal space	
H'1C000000 to H'1FFFFFFF	CS5	Normal space	
H'20000000 to H'23FFFFFF	CS0	Normal space	Cache-disabled
H'24000000 to H'27FFFFFF	CS1	Normal space	
H'28000000 to H'2BFFFFFF	SDRAM0	SDRAM	
H'2C000000 to H'2FFFFFFF	SDRAM1	SDRAM	
H'30000000 to H'33FFFFFF	CS2	Normal space	
H'34000000 to H'37FFFFFF	CS3	Normal space	
H'38000000 to H'3BFFFFFF	CS4	Normal space	
H'3C000000 to H'3FFFFFFF	CS5	Normal space	

Internal Address	Space	Memory to be Connected	Cache
H'40000000 to H'4FFFFFFF	CS6	Normal space	Cache-disabled
H'50000000 to H'E7FFFFFF	Other	Reserved area*	—
H'E8000000 to H'EFFFFFFF	Other	On-chip peripheral modules, reserved area*	—
H'F0000000 to H'FF3FFFFF	Other	Cache address array space, reserved area*	—
H'FF400000 to H'FFF7FFFF	Other	On-chip peripheral modules, reserved area*	—
H'FFF80000 to H'FFFBFFFF	Other	On-chip RAM, reserved area*	—
H'FFFC0000 to H'FFFFFFFF	Other	On-chip peripheral modules, reserved area*	—

Note: * For the on-chip RAM space, access the addresses shown in section 24, On-Chip RAM. For the on-chip peripheral module space, access the addresses shown in section 28, List of Registers. Do not access addresses which are not described in these sections. Otherwise, correct operation cannot be guaranteed.

9.3.2 Data Bus Width and Pin Function Setting for Individual Areas

In this LSI the data bus width of area 0 can be set to 8, 16, or 32 bits through external pins during a power-on reset. The data bus widths of areas 1 to 6 can be modified through register settings during program execution. Note that the selectable data bus widths may be limited depending on the connected memory type.

After a power-on reset, the LSI starts execution of the program stored in the external memory allocated in area 0.

For details on pin function settings, see section 23, Pin Function Controller (PFC).

Table 9.3 Correspondence between External Pin (MD1 and MD0) Settings and Data Bus Width

MD1	MD0	Data Bus Width
1	1	32 bits
	0	16 bits
0	1	8 bits
	0	Reserved (setting prohibited)

9.4 Register Descriptions

The BSC has the following registers.

All registers are initialized by a power-on reset or in deep standby mode.

Do not access spaces other than area 0 until settings are completed for the connected memory interface.

Table 9.4 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
CS0 control register	CS0CNT	R/W	H'00010000/ H'00110000/ H'00210000*	H'FF420000	8, 16, 32
CS0 recovery cycle setting register	CS0REC	R/W	H'00000000	H'FF420008	8, 16, 32
CS1 control register	CS1CNT	R/W	H'00000000	H'FF420010	8, 16, 32
CS1 recovery cycle setting register	CS1REC	R/W	H'00000000	H'FF420018	8, 16, 32
CS2 control register	CS2CNT	R/W	H'00000000	H'FF420020	8, 16, 32
CS2 recovery cycle setting register	CS2REC	R/W	H'00000000	H'FF420028	8, 16, 32
CS3 control register	CS3CNT	R/W	H'00000000	H'FF420030	8, 16, 32
CS3 recovery cycle setting register	CS3REC	R/W	H'00000000	H'FF420038	8, 16, 32
CS4 control register	CS4CNT	R/W	H'00000000	H'FF420040	8, 16, 32
CS4 recovery cycle setting register	CS4REC	R/W	H'00000000	H'FF420048	8, 16, 32
CS5 control register	CS5CNT	R/W	H'00000000	H'FF420050	8, 16, 32
CS5 recovery cycle setting register	CS5REC	R/W	H'00000000	H'FF420058	8, 16, 32
CS6 control register	CS6CNT	R/W	H'00000000	H'FF420060	8, 16, 32
CS6 recovery cycle setting register	CS6REC	R/W	H'00000000	H'FF420068	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
SDRAMC0 control register	SDC0CNT	R/W	H'00000000	H'FF420100	8, 16, 32
SDRAMC1 control register	SDC1CNT	R/W	H'00000000	H'FF420110	8, 16, 32
CS0 mode register	CSMOD0	R/W	H'00000000	H'FF421000	8, 16, 32
CS0 wait control register 1	CS1WCNT0	R/W	H'1F1F0707	H'FF421004	8, 16, 32
CS0 wait control register 2	CS2WCNT0	R/W	H'00000007	H'FF421008	8, 16, 32
CS1 mode register	CSMOD1	R/W	H'00000000	H'FF421010	8, 16, 32
CS1 wait control register 1	CS1WCNT1	R/W	H'1F1F0707	H'FF421014	8, 16, 32
CS1 wait control register 2	CS2WCNT1	R/W	H'00000007	H'FF421018	8, 16, 32
CS2 mode register	CSMOD2	R/W	H'00000000	H'FF421020	8, 16, 32
CS2 wait control register 1	CS1WCNT2	R/W	H'1F1F0707	H'FF421024	8, 16, 32
CS2 wait control register 2	CS2WCNT2	R/W	H'00000007	H'FF421028	8, 16, 32
CS3 mode register	CSMOD3	R/W	H'00000000	H'FF421030	8, 16, 32
CS3 wait control register 1	CS1WCNT3	R/W	H'1F1F0707	H'FF421034	8, 16, 32
CS3 wait control register 2	CS2WCNT3	R/W	H'00000007	H'FF421038	8, 16, 32
CS4 mode register	CSMOD4	R/W	H'00000000	H'FF421040	8, 16, 32
CS4 wait control register 1	CS1WCNT4	R/W	H'1F1F0707	H'FF421044	8, 16, 32
CS4 wait control register 2	CS2WCNT4	R/W	H'00000007	H'FF421048	8, 16, 32
CS5 mode register	CSMOD5	R/W	H'00000000	H'FF421050	8, 16, 32
CS5 wait control register 1	CS1WCNT5	R/W	H'1F1F0707	H'FF421054	8, 16, 32
CS5 wait control register 2	CS2WCNT5	R/W	H'00000007	H'FF421058	8, 16, 32
CS6 mode register	CSMOD6	R/W	H'00000000	H'FF421060	8, 16, 32
CS6 wait control register 1	CS1WCNT6	R/W	H'1F1F0707	H'FF421064	8, 16, 32
CS6 wait control register 2	CS2WCNT6	R/W	H'00000007	H'FF421068	8, 16, 32
SDRAM refresh control register 0	SDRFCNT0	R/W	H'00000000	H'FF422000	8, 16, 32
SDRAM refresh control register 1	SDRFCNT1	R/W	H'0000xxxx	H'FF422004	16, 32
SDRAM initialization register 0	SDIR0	R/W	H'00000xxx	H'FF422008	8, 16, 32
SDRAM initialization register 1	SDIR1	R/W	H'00000000	H'FF42200C	8, 16, 32

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
SDRAM power-down control register	SDPWDCNT	R/W	H'00000000	H'FF422010	8, 16, 32
SDRAM deep-power-down control register	SDDPWDCNT	R/W	H'00000000	H'FF422014	8, 16, 32
SDRAM0 address register	SD0ADR	R/W	H'00000x0x	H'FF422020	8, 16, 32
SDRAM0 timing register	SD0TR	R/W	H'000xxx0x	H'FF422024	8, 16, 32
SDRAM0 mode register	SD0MOD	R/W	H'0000xxxx	H'FF422028	16, 32
SDRAM1 address register	SD1ADR	R/W	H'00000x0x	H'FF422040	8, 16, 32
SDRAM1 timing register	SD1TR	R/W	H'000xxx0x	H'FF422044	8, 16, 32
SDRAM1 mode register	SD1MOD	R/W	H'0000xxxx	H'FF422048	16, 32
SDRAM status register	SDSTR	R/W	H'00000000	H'FF4220E4	8, 16, 32
SDRAM clock stop control signal setting register	SDCKSCNT	R/W	H'0000000F	H'FF4220E8	8, 16, 32
AC characteristics switching register	ACSWR	R/W	H'00000000	H'FFFD8808	8, 16, 32

Note: * Depends on the setting of the MD pin.

9.4.1 CSn Control Register (CSnCNT) (n = 0 to 6)

CSnCNT selects the width of the external bus and controls the operation of the CSC interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	BSIZE[1:0]		—	—	—	EXENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0*1	0*1	0	0	0	0*2
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
21, 20	BSIZE[1:0]	00*[1]	R/W	External Bus Width Select These bits specify the width of the data bus for the external device of the corresponding channel of CSC. The initial value for the data bus width for CSC channel 0 (CS0) differs depending on the settings of pins MD1 and MD0. 10: 8-bit bus 00: 16-bit bus 01: 32-bit bus
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	EXENB	0*[2]	R/W	Operation Enable This bit enables or disables the operation for the corresponding channel of CSC. The initial value corresponding to CS0 only is operation enabled (EXENB = 1). 0: Operation disabled 1: Operation enabled
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. The initial value of the BSIZE bits in CS0 differs depending on the settings of pins MD1 and MD0.
2. The initial value of the EXENB bit in CS0 is 1.

To disable the operation for each channel, forcibly write out data tentatively stored in internal write buffer. The procedure is as follows:

1. Execute read access to the channel whose operation is to be disabled.
2. Then, write 0 to the EXENB bit (operation disabled).

9.4.2 CSn Recovery Cycle Setting Register (CSnREC) (n = 0 to 6)

CSnREC specifies the number of data recovery cycles to be inserted after read or write accesses.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 24	WRCV[3:0]	0000	R/W	Post-Write Data Recovery Cycle Setting These bits specify the number of data recovery cycles to be inserted after write accesses to the external bus. If a value other than 0 is selected, between 1 and 15 data recovery cycles are inserted when a write access to the external bus is followed by a read access to the external bus. (Data recovery cycles are inserted even when access is performed sequentially to the same CSC channel.) Note that if idle cycles occur between accesses to the external bus, the number of data recovery cycles inserted is reduced by the number of idle cycles. 0000: 0 cycle 0001: 1 cycles : 1111: 15 cycles
23 to 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	Description
19 to 16	RRCV[3:0]	0000	R/W	Post-Read Data Recovery Cycle Setting These bits specify the number of data recovery cycles to be inserted after read accesses to the external bus. If a value other than 0 is selected, data recovery cycles are inserted in the following cases: If a read access to the external bus is followed by a write access to the external bus. (Data recovery cycles are inserted even when access is performed sequentially to the same CSC channel.) If a read access to the external bus is followed by a read access to a different CSC channel. (No data recovery cycles are inserted in cases of sequential read accesses to the same CSC channel.) Note that if idle cycles occur between accesses to the external bus, the number of data recovery cycles inserted is reduced by the number of idle cycles. 0000: 0 cycle 0001: 1 cycles : 1111: 15 cycles
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. When accessing SDRAM, there is no danger of data collision on the bus due to timing. Consequently, there is no data recovery cycle setting for SDRAM. (The value is fixed at 0 cycles.)

2. Writing to the CSn recovery cycle setting register (CSnREC) must be done while CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) can be enabled by setting EXENB = 1. To enable channel 0, stop the DMAC and set EXENB to 1 between the reset release and data write access to CS0.

9.4.3 SDRAMCm Control Register (SDCmCNT) (m = 0, 1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	BSIZE[1:0]		—	—	—	EXENB
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	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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	BSIZE[1:0]	00	R/W	External Bus Width Select These bits specify the width of the data bus for the external device of the corresponding channel of CSC. 10: 8-bit bus 00: 16-bit bus 01: 32-bit bus
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	EXENB	0	R/W	Operation Enable This bit enables or disables the operation for the corresponding channel of CSC. 0: Operation disabled 1: Operation enabled
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

To disable the operation for each channel, forcibly write out data tentatively stored in internal write buffer. The procedure is as follows:

1. Execute read access to the channel whose operation is to be disabled.
2. Then, write 0 to the EXENB bit (operation disabled).

9.4.4 CSn Mode Register (CSMODn) (n = 0 to 6)

CSMODn selects the mode for page read access and the bit boundary for page access, enables page read/write access and external wait, and selects the mode for write access.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PR MOD	—	PBCNT[1:0]		—	—	PW ENB	PR ENB	—	—	—	—	EW ENB	—	—	WR MOD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R/W	R	R	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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	PRMOD	0	R/W	Page Read Access Mode Select This bit selects the operating mode for page read access. Clearing PRMOD to 0 selects the normal access compatible mode. In this mode the $\overline{RD}$ signal is negated each time a unit of data is read and an RD assert wait is inserted. Setting PRMOD to 1 selects the external data read sequential assert mode. In this mode $\overline{RD}$ is asserted continuously between page accesses. 0: Normal access compatible mode 1: External data read sequential assert mode
30	—	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
29, 28	PBCNT[1:0]	00	R/W	Page Access Bit Boundary Select These bits select the bit boundary for page access operation. When the bit boundary specified by PBCNT is exceeded during page access, page access operation is halted temporarily (the $\overline{CSn}$ signal is negated), and then page access operation begins again. The value written to these bits is valid only when either of the PWENB bit or the PRENB bit is set to 1. 00: 64-bit boundary 01: 128-bit boundary 10: 256-bit boundary 11: Setting prohibited
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25	PWENB	0	R/W	Page Write Access Enable This bit is used to enable page write access. 0: Page write access disabled 1: Page write access enabled
24	PRENB	0	R/W	Page Read Access Enable This bit is used to enable page read access. 0: Page write access disabled 1: Page write access enabled
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19	EWENB	0	R/W	External Wait Enable This bit is used to enable or disable external wait input. When EWENB is set to 1, external wait input is enabled and the number of wait states per cycle can be controlled using the external wait signal ($\overline{WAIT}$). In this case wait cycles are inserted while the $\overline{WAIT}$ signal is low level. When EWENB is cleared to 0, the $\overline{WAIT}$ signal is invalid. 0: External wait disabled 1: External wait enabled

Bit	Bit Name	Initial Value	R/W	Description
18, 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	WRMOD	0	R/W	Write Access Mode Select This bit selects the operating mode for write access. Clearing WRMOD to 0 selects the byte-write strobe mode. In this mode data writes are controlled by multiple write signals ($\overline{WR3}$ to $\overline{WR0}$) that correspond to the individual byte positions. Setting WRMOD to 1 selects the one-write strobe mode. In this mode, data writes are controlled by multiple byte control signals ($\overline{BC3}$ to $\overline{BC0}$) that correspond to the individual byte positions and a single write signal ($\overline{WR0}$ for the 8-bit bus width channel, $\overline{WR1}$ for the 16-bit bus width channel, and $\overline{WR3}$ for the 32-bit bus width channel) 0: Byte-write strobe mode 1: One-write strobe mode
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Writing to the CSn mode register (CSMODn) must be done while CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) can be enabled by setting EXENB = 1. To enable channel 0, stop the DMAC and set EXENB to 1 between the reset release and data write access to CS0.

9.4.5 CSn Wait Control Register 1 (CS1WCNTn) (n = 0 to 6)

CS1WCNTn specifies the number of wait states inserted into the read/write cycle or page read/page write cycle.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Initial value:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28 to 24	CSRWAIT [4:0]	11111	R/W	Read Cycle Wait Select These bits specify the number of wait states inserted into the initial normal read cycle and page read cycle. 00000: 0 wait states : 11111: 31 wait states
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	CSWWAIT [4:0]	11111	R/W	Write Cycle Wait Select These bits specify the number of wait states inserted into the initial normal write cycle and page write cycle. 00000: 0 wait states : 11111: 31 wait states
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 8	CSPRWAIT [2:0]	111	R/W	Page Read Cycle Wait Select These bits specify the number of wait states inserted into the second and subsequent page read cycles. This setting is valid when the page read access enable bit (PRENB) is set to 1. 000: 0 wait state : 111: 7 wait states
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CSPWWAIT [2:0]	111	R/W	Page Write Cycle Wait Select These bits specify the number of wait states inserted into the second and subsequent page write cycles. This setting is valid when the page write access enable bit (PWENB) is set to 1. 000: 0 wait state : 111: 7 wait states

Notes: 1. Make sure the page read and page write cycle wait select (CSPRWAIT and CSPWWAIT) settings are within the range defined by the read and write cycle wait select (CSRWAIT and CSWWAIT) settings. Select each wait cycle number according the system configuration incorporated.

2. Writing to the CSn wait control register 1 (CS1WCNTn) must be done while CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) can be enabled by setting EXENB = 1. To enable channel 0, stop the DMAC and set EXENB to 1 between the reset release and data write access to CS0.

9.4.6 CSn Wait Control Register 2 (CS2WCNTn) (n = 0 to 6)

CS2WCNTn specifies the number of wait states and the number of delay cycles.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/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
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 28	CSON [2:0]	000	R/W	CS Assert Wait Select These bits specify the number of wait states inserted before the external chip select signal ($\overline{CSn}$) is asserted. 000: 0 wait state : 111: 7 wait states
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 24	WDON [2:0]	000	R/W	Write Data Output Wait Select These bits specify the number of wait states inserted before data is output to the external data bus. 000: 0 wait state : 111: 7 wait states
23	—	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
22 to 20	WRON [2:0]	000	R/W	WR Assert Wait Select These bits specify the number of wait states inserted before the external data write signal ($\overline{WR3}$ to $\overline{WR0}$) is asserted. 000: 0 wait state : 111: 7 wait states
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
18 to 16	RDON [2:0]	000	R/W	RD Assert Wait Select These bits specify the number of wait states inserted before the external data read signal ($\overline{RD}$) is asserted. 000: 0 wait state : 111: 7 wait states
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	WDOFF [2:0]	000	R/W	Write Data Output Delay Cycle Select These bits specify the number of cycles from the end of the wait cycle during write operation (negation of the $\overline{WR3}$ to $\overline{WR0}$ signals) and the negation of the external data bus. 000: 0 wait state : 111: 7 wait states
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	CSWOFF [2:0]	000	R/W	Write Operation CS Delay Cycle Select These bits specify the number of cycles from the end of the wait cycle during write access operation (negation of the $\overline{WR3}$ to $\overline{WR0}$ signals) and the negation of the $\overline{CS6}$ to $\overline{CS0}$ signal. 000: 0 wait state : 111: 7 wait states
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	CSROFF [2:0]	111	R/W	Read Operation CS Delay Cycle Select These bits specify the number of cycles from the end of the wait cycle during read access operation (negation of the $\overline{RD}$ signal) and the negation of the $\overline{CS6}$ to $\overline{CS0}$ signal. 000: 0 wait state : 111: 7 wait states

Notes: 1. Select each wait cycle number or extended cycle number according the system configuration incorporated.

2. Writing to the CSn wait control register 2 (CS2WCNTn) must be done while CSC for the corresponding channel is disabled (EXENB = 0). Only channel 0 (CS0) can be enabled by setting EXENB = 1. To enable channel 0, stop the DMAC and set EXENB to 1 between the reset release and data write access to CS0.

3. Each bit must be set under the following restrictions.
   - When page access is disabled (PRENB, PWENB = 0)
     CSON ≤ min (CSRWAIT, CSWWAIT), WDON ≤ CSWWAIT
     WRON ≤ CSWWAIT, RDON ≤ CSRWAIT
     WDOFF ≤ CSWOFF
   - When page access is enabled (PRENB = 1 or PWENB = 1)
     In addition to the restrictions for disabled page access case, the following restrictions are required.
     CSON ≤ min (CSPRWAIT, CSPWWAIT)
     WRON ≤ CSPWWAIT, RDON ≤ CSPRWAIT
     WDON ≤ CSPWWAIT

9.4.7 SDRAM Refresh Control Register 0 (SDRFCNT0)

SDRFCNT0 controls self-refresh operation.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DSFEN	0	R/W	SDRAM Common Self-Refresh Operation Enable This bit controls self-refresh operation for all channels simultaneously. Setting DSFEN to 1 performs auto-refresh cycle operation, immediately after which self-refresh operation begins. Clearing DSFEN to 0 ends self-refresh operation, and auto-refresh operation resumes immediately afterward. The value written to this bit is reflected when self-refresh operation starts, if DSFEN was set to 1, or when auto-refresh operation starts following the end of self-refresh operation, if DSFEN was cleared to 0. 0: Self-refresh disabled 1: Self-refresh enabled

9.4.8 SDRAM Refresh Control Register 1 (SDRFCNT1)

SDRFCNT1 controls auto-refresh operation.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DREFW[3:0]				DRFC[11:0]											
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 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	DRFEN	0	R/W	Auto-Refresh Operation Enable This bit controls auto-refresh operation for all channels simultaneously. When DRFEN is cleared to 0, auto-refresh operation does not take place. Auto-refresh operates when DRFEN is set to 1. Clearing this bit to 0 while auto-refresh is enabled causes DRFEN to be cleared to 0, and auto-refresh operation to halt, after the end of the next auto-refresh cycle. Setting this bit to 1 while auto-refresh is enabled causes auto-refresh operation to commence as soon as DRFEN is set to 1, and refresh requests are then generated at fixed intervals determined by a counter. The interval at which refresh requests are generated is determined by the set value of the auto-refresh request interval setting (DRFC) bits. Refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. If a SDRAM access and refresh request are generated at the same time, the refresh request takes precedence. 0: Auto-refresh disabled 1: Auto-refresh enabled

Bit	Bit Name	Initial Value	R/W	Description
15 to 12	DREFW [3:0]	Undefined	R/W	Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting These bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles. The DREFW bits can be written to at any time, regardless of the state of the auto-refresh operation enable (DRFEN) bit. If auto-refresh is disabled, the value written to these bits takes effect immediately. If auto-refresh is enabled, the value written to these bits takes effect immediately if an auto-refresh cycle is not in progress. If an auto-refresh cycle is in progress, the new value takes effect after the cycle completes. 0000: 1 cycle 0001: 2 cycles 0010: 3 cycles : 1111: 16 cycles
11 to 0	DRFC [11:0]	Undefined	R/W	Auto-Refresh Request Interval Setting These bits specify the auto-refresh interval. The DRFC bits can be written to at any time, regardless of the state of the auto-refresh operation enable (DRFEN) bit. If auto-refresh is disabled, the value written to these bits takes effect immediately. If auto-refresh is enabled, the value written to these bits is reflected in the operation of the refresh counter from the next auto-refresh request generated. 000000000000: Setting prohibited 000000000001: 2 cycles 000000000010: 3 cycles : 111111111111: 4096 cycles

Note: Auto-refresh requests are not accepted while multiple read or write accesses are in progress, or during a transfer using DMAC, so the auto-refresh interval may become enlarged in some cases. Set the DRFC bits to an auto-refresh request interval value that satisfies the auto-refresh interval specification of the SDRAM being used. Furthermore, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle.

Auto-Refresh Request Interval and DRFC Set Value:

SDRAMC includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. The following equation is used to calculate the set value for the DRFC bits from the auto-refresh request interval.

DRFC = (Auto-refresh request interval / Bus clock cycle) – 1

Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. However, the counter value is updated regardless or whether or not the request was accepted. Note that if two or more auto-refresh requests are generated while SDRAM is being accessed, the second and subsequent requests are ignored.

9.4.9 SDRAM Initialization Register 0 (SDIR0)

SDIR0 specifies the SDRAM initialization sequence timing.

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
	—	—	—	—	—	DPC[2:0]			DARFC[3:0]				DARFI[3:0]			
Initial value:	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	DPC[2:0]	Undefined	R/W	Initialization Precharge Cycle Count Setting These bits specify the number of precharge cycles in the SDRAM initialization sequence. 000: 3 cycles 001: 4 cycles : 111: 10 cycles

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	DARFC [3:0]	Undefined	R/W	Initialization Auto-Refresh Count These bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence. 0000: Setting prohibited 0001: 1 time : 1111: 15 times
3 to 0	DARFI[3:0]	Undefined	R/W	Initialization Auto-Refresh Interval These bits specify the interval at which auto-refresh commands are issued in the SDRAM initialization sequence. 0000: 3 cycles 0001: 4 cycles 0010: 5 cycles : 1111: 18 cycles

Note: Make settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

9.4.10 SDRAM Initialization Register 1 (SDIR1)

SDIR1 controls activation of the SDRAM initialization sequence.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	DINIST	0	R/W	Initialization Status When set to 1, this bit indicates that an SDRAM initialization sequence is in progress for channel SDRAM0 or SDRAM1. 0: Initialization sequence not progress 1: Initialization sequence in progress
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DINIRQ	0	R/W	Common Initialization Sequence Start Setting this bit to 1 causes the SDRAM initialization sequence to start and automatically sets the initialization status bit (DINIST) to 1. The initialization status bit (DINIST) is cleared automatically after the initialization sequence ends. The value written to the DINIRQ bit is not retained. 0: Invalid 1: Initialization sequence start

9.4.11 SDRAM Power-Down Control Register (SDPWDCNT)

SDPWDCNT controls transition to and recovery from power-down 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPWD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DPWD	0	R/W	SDRAM Common Power-Down Enable This bit controls transition to and recovery from power-down mode for all channels simultaneously. Setting DPWD to 1 causes all channels to transition to power-down mode. Clearing DPWD to 0 causes all channels to recover from power-down mode. If an auto-refresh is in progress, the transition to power-down mode is delayed until the auto-refresh completes. 0: Power-down disabled 1: Power-down enabled

9.4.12 SDRAM Deep-Power-Down Control Register (SDDPWDCNT)

SDDPWDCNT controls transition to and recovery from deep-power-down 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDPD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DDPD	0	R/W	SDRAM Common Deep-Power-Down Enable This bit controls transition to and recovery from deep-power-down mode for all channels simultaneously. Setting DDPD to 1 causes all SDRAM channels to transition to deep-power-down mode. Clearing DDPD to 0 causes all SDRAM channels to recover from deep-power-down mode. If an auto-refresh is in progress, the transition to deep-power-down mode is delayed until the auto-refresh completes. 0: Deep-power-down disabled 1: Deep-power-down enabled

9.4.13 SDRAMm Address Register (SDmADR) (m = 0, 1)

SDmADR specifies the data bus width and the channel size of SDRAM.

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
	—	—	—	—	—	—	DDBW[1:0]		—	—	—	—	—	DSZ[2:0]		
Initial value:	0	0	0	0	0	0	—	—	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	DDBW[1:0]	Undefined	R/W	SDRAM Data Bit Width Setting These bits specify the width of the SDRAM bus. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DSZ[2:0]	Undefined	R/W	Channel Size Setting These bits specify the size of channels 0 and 1. If a size smaller than SDRAM area 0 or 1 is selected, ghost memory will result. When accessing 32-bit data in SDRAM with a 16-bit bus width, the 16 bits of the first half of the address (A1 = 0) are accessed first, and then the 16 bits of the second half of the address (A1 = 1) are accessed.

9.4.14 SDRAMm Timing Register (SDmTR) (m = 0, 1)

SDmTR specifies the timing for read and write accesses to SDRAM.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DRCD[1:0]		DPCG[2:0]			DWR	—	—	—	—	—	DCL[2:0]		
Initial value:	0	0	—	—	—	—	—	—	0	0	0	0	0	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	DRAS[2:0]	Undefined	R/W	Row Active Interval Setting These bits specify the minimum interval that must elapse between the SDRAM row activation command (ACT) and deactivation (PRA). 000: 1 cycle : 111: 8 cycles
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	DRCD[1:0]	Undefined	R/W	Row Column Latency Setting These bits specify the SDRAM row column latency. 00: 1 cycles 01: 2 cycles 10: 3 cycles 11: 4 cycles

Bit	Bit Name	Initial Value	R/W	Description
11 to 9	DPCG[2:0]	Undefined	R/W	Row Precharge Interval Setting These bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PRA) and the next valid command. 000: 1 cycles : 111: 8 cycles
8	DWR	0	R/W	Write Recovery Interval Setting This bit specifies the minimum interval that must elapse between the SDRAM write command (WRITE) and deactivation (PRA). 0: 1 cycles 1: 2 cycles
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DCL[2:0]	Undefined	R/W	SDRAM Controller Column Latency Setting These bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM it is necessary to use the separate SDRAMm mode register (SDmMOD), which is described below. 000: Setting prohibited 001: 1 cycles 010: 2 cycles 011: 3 cycles 1xx: Setting prohibited

[Legend]

x: Don't care

9.4.15 SDRAMm Mode Register (SDmMOD) (m = 0, 1)

SDmMOD specifies the values to be written to the SDRAM mode register or extended mode register. Writing to this register causes a mode register set command or extended mode register set command to be issued automatically to SDRAM.

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
	—	DMR[14:0]														
Initial value:	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
31 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14 to 0	DMR[14:0]	Undefined	R/W	Mode Register Setting Writing to these bits causes a mode register set command or extended mode register set command to be issued to SDRAM. The setting of the DMR bits is output as A16 to A2 signals. The distinction between the mode register set command and extended mode register set command is made on the bases of the SDRAM bank address. Write operation: A mode register set command is issued. DMR bit: b14 b13 ... b0 ↓ ↓ ↓ A16 to A2 signal: A16 A15 ... A2

Notes: The following points should be kept in mind regarding SDRAMm mode register settings.

1. Make sure to set a burst length of 1 for SDRAM. Operation cannot be guaranteed with settings other than burst length 1.
2. The SDRAM column latency must match the setting of the SDRAM controller column latency setting bits (DCL) in SDRAMC. Operation cannot be guaranteed if the latency settings do not agree.
3. Check to make sure the status bits (DSRFST, DPWDST, DDPDST, and DMRSST) in the SDRAM status register (SDSTR) are all cleared to 0.

9.4.16 SDRAM Status Register (SDSTR)

SDSTR consists of the status flags that indicate the status of operation during self-refresh, initialization sequences, power-down mode, deep-power-down mode, and mode register setting.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DSRFST	DINIST	DPWDST	DDPDST	DMRSST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	DSRFST	0	R	Self-Refresh Transition/Recovery Status When set to 1, this bit indicates that a transition to or recovery from self-refresh operation is in progress for channel SDRAM0 or SDRAM1. 0: Transition/recovery not in progress 1: Transition/recovery in progress
3	DINIST	0	R	Initialization Status When set to 1, this bit indicates that an initialization sequence is in progress for channel SDRAM0 or SDRAM1. This bit has the same function as the DINIST bit in SDIR1. 0: Initialization sequence not in progress 1: Initialization sequence in progress

Bit	Bit Name	Initial Value	R/W	Description
2	DPWDST	0	R	Power-Down Transition/Recovery Status When set to 1, this bit indicates that a transition to or recovery from power-down mode is in progress for a channel from SDRAM0 to SDRAM3. 0: Initialization sequence not in progress 1: Initialization sequence in progress
1	DDPDST	0	R	Deep-Power-Down Transition/Recovery Status When set to 1, this bit indicates that a transition to or recovery from deep-power-down mode is in progress for channel SDRAM0 or SDRAM1. 0: Transition/recovery not in progress 1: Transition/recovery in progress
0	DMRSST	0	R	Mode Register Setting Status When set to 1, this bit indicates that mode register setting is in progress for channel SDRAM0 or SDRAM1. 0: Mode register setting not in progress 1: Mode register setting in progress

"Transition to or recovery from in progress" refers to the interval from the point at which the bits listed in table 9.5 are written to until the corresponding commands are issued.

Table 9.5 List of Status Registers and Bits Requiring Checking

Function	Register	Bits
Self-refresh	SDRFCNT0	DSFENCm, DSFEN
Initialization sequence	SDIR1	DINIRQCm, DINIRQ
Power-down	SDPWDCNT	DPWDCm, DPWD
Deep-power-down	SDDPDCNT	DDPDCm, DDPD
Mode register setting	SDmMOD	DMR

Note: Execution of a self-refresh, a transition to or recovery from power-down or deep-power-down mode, an initialization sequence, or mode register setting may only be performed when all status bits are cleared to 0. Do not rewrite the registers (bits) listed below when any of the status bits (DSRFST, DINIST, DPWDST, DDPDST, DMRSST) is set to 1.

9.4.17 SDRAM Clock Stop Control Signal Setting Register (SDCKSCNT)

SDCKSCNT enables or disables the clock stop control signal (internal signal in the chip) and specifies the number of assert cycles.

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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DCKSC[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/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 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	DCKSEN	0	R/W	Clock Stop Control Signal Enable This bit is used to enable or disable the clock stop control signal. When enabled, the clock stop control signal operates during transition to and from deep-power-down mode and stops the CKIO (high level). When disabled, the clock stop control signal stays low level. 0: Clock stop control signal disabled 1: Clock stop control signal enabled
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 to 0	DCKSC [7:0]	H'0F	R/W	Clock Stop Control Signal Assert Cycle Count Setting These bits specify the interval from the point at which the deep-power-down transition command is issued until the clock stop signal goes high level to stop the CKIO (high level), and the interval from the point at which the clock stop signal goes low level to start the CKIO operation until the recover command is issued. 00000000: 0 cycle : 00001111: 15 cycles : 11111111: 255 cycles

9.4.18 AC Characteristics Switching Register (ACSWR)

When writing to the external address space or making SDRAM settings in power-on reset exception handling or cancellation of deep standby mode, be sure to set bits ACOSW[3:0] in ACSWR to B'0011 beforehand.

ACSWR is initialized to H'00000000 by a power-on reset and entry to deep standby mode, but is not initialized by a manual reset, entry to sleep mode, or entry to 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/W	R/W	R/W	R/W	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
	—	—	—	—	—	—	—	—	—	—	—	—	ACOSW[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/W	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 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	ACOSW[3:0]	0000	R/W	AC Characteristics Switch These bits specify AC characteristics switching. 0000: Does not extend the delay time 0011: Switches characteristics and extends the delay time Other than above: Setting prohibited

9.5 Operation

9.5.1 CSC Interface

(1) Normal Access

Normal read/write operation is used for all bus access when page read/write access is disabled (PRENB = 0, PWENB = 0). Even when page read/write access is enabled (PRENB = 1, PWENB = 1), normal read/write operation is employed in cases where page access cannot be used. Figure 9.2 shows the basic operation of the external bus control signals in read operation, and figure 9.3 shows the basic operation of these signals in write operation.

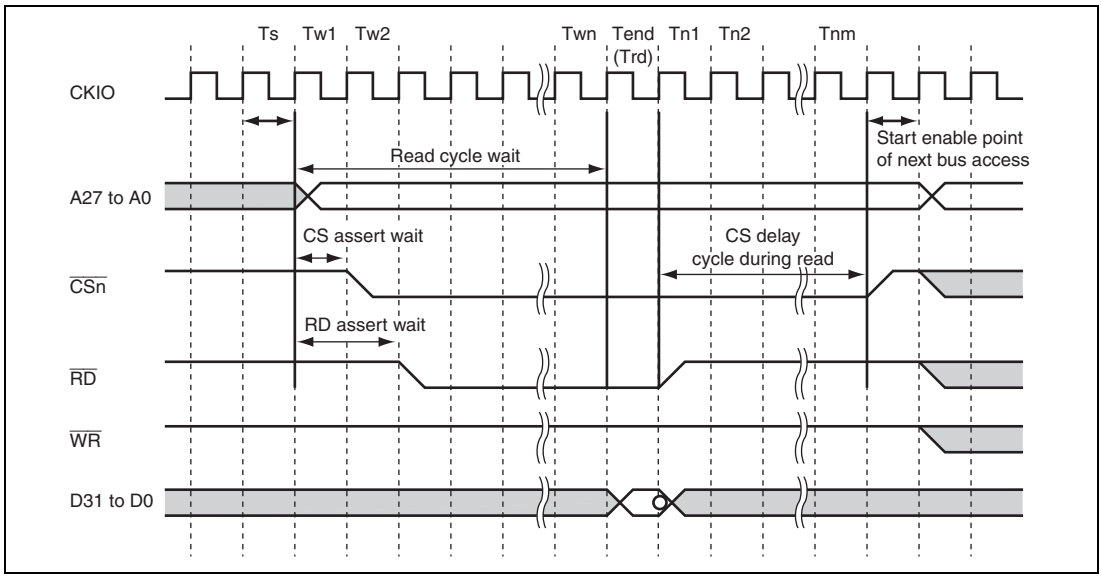


Figure 9.2 Basic Bus Timing (Read Operation)

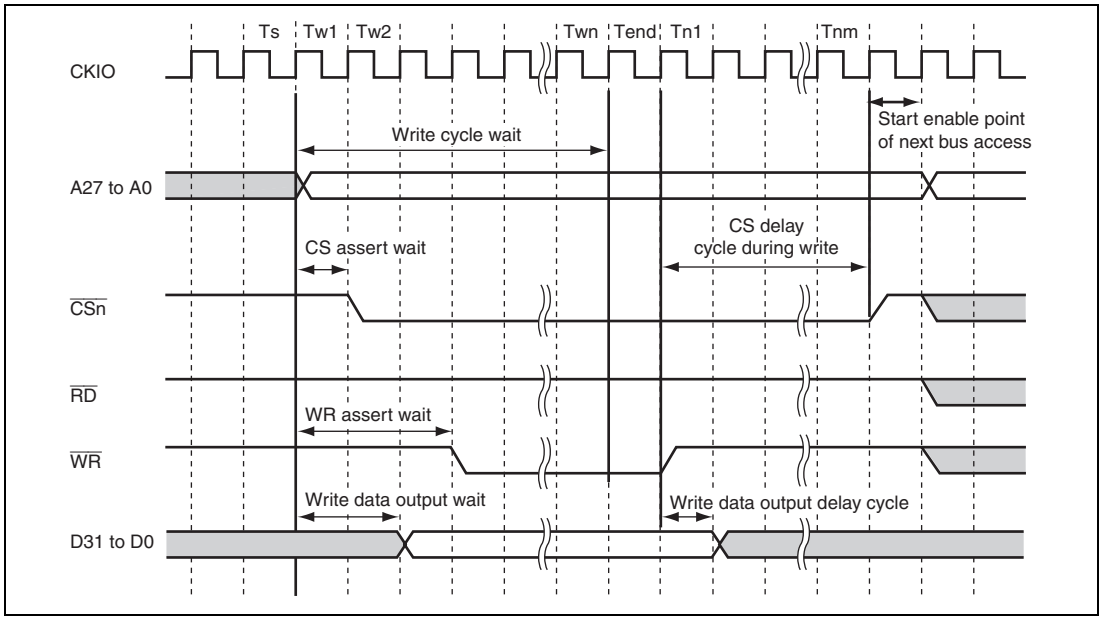


Figure 9.3 Basic Bus Timing (Write Operation)

1. Ts (Internal Bus Access Start)

   This is a bus access request cycle initiated by the internal bus master and with the external bus as the target. $\overline{CSn}$ is always high during this cycle. In the next cycle A27 to A0 and the write data change.

2. Tw1 to Twn (Read Cycle Wait, Write Cycle Wait)

   These are the cycles between internal bus access start and the wait end cycle. A duration of from 0 to 31 clocks may be selected. During this interval the $\overline{CSn}$, $\overline{RD}$, and $\overline{WR}$ control signals are asserted (low level) in accordance with the wait settings. The assert timing can be controlled using the CS assert wait, RD assert wait, WR assert wait, and write data output wait bits in CSn control registers 1 and 2. The number of wait cycles can be set to from 0 to 7 clocks, with the count starting from the cycle following internal bus access start (Ts). The number of clocks selected must be no greater than the number of read/write cycle wait cycles.

3. Tend (Wait End Cycle)

   This is the final cycle in a series of read cycle wait or write cycle wait cycles. The $\overline{RD}$ or $\overline{WR}$ signal is negated (high level) in the next cycle.

4. Tn1 to Tnm (CS Delay Cycle)

   These are the cycles between the wait end cycle and when $\overline{\text{CSn}}$ is negated (high level). The negation timing can be controlled using write data output delay cycles. The number of cycles is counted beginning from the wait end cycle. In write access or if the number of CS delay cycles during a read is other than 0 or 1, the succeeding bus access can start from the cycle following the CS delay cycle end. If the number of CS delay cycles is 0 or 1 in read access, the succeeding bus access can start after the end of the read data sample cycle (see below).
5. Trd (Read Data Sample Cycle)

   This is the sample cycle for read data.

(2) Page Access

Page read and write operation is employed for bus accesses for which page access can be used if page write access enable (PWENB = 1) and page read access enable (PRENB = 1) have been selected. Page access is used in the following cases.

1. CPU burst access (cache replacement)
2. When longword (32-bit) access to an 8-bit or 16-bit external data bus has been performed
3. When word (16-bit) access to an 8-bit external data bus has been performed

Table 9.6 shows the way addresses are modified in cases 1 above.

Table 9.6 Address Modification during Burst Access

Bus Master	Burst Mode	Address Modification
CPU	Increment	Incremented by single transfer byte count only.

Note: Wrap boundary: Single transfer byte count × Burst transfer length

Figure 9.4 shows the basic operation of the external bus control signals in page read operation, and figure 9.5 shows the basic operation of these signals in write operation. Note that if the number of data bits accessed in a single burst is greater than the single page access bit boundary setting of the PBCNT bits in the mode register, a single burst access will trigger multiple page accesses. Regardless of whether the bust mode is increment or wraparound, page access stops temporarily (the $\overline{\text{CSn}}$ signal is negated) at the point when the address exceeds the page boundary, and page access operation starts again. If the number of data bits accessed in a single burst is smaller than the page boundary bit count, a single page access is sufficient to complete the burst transfer.

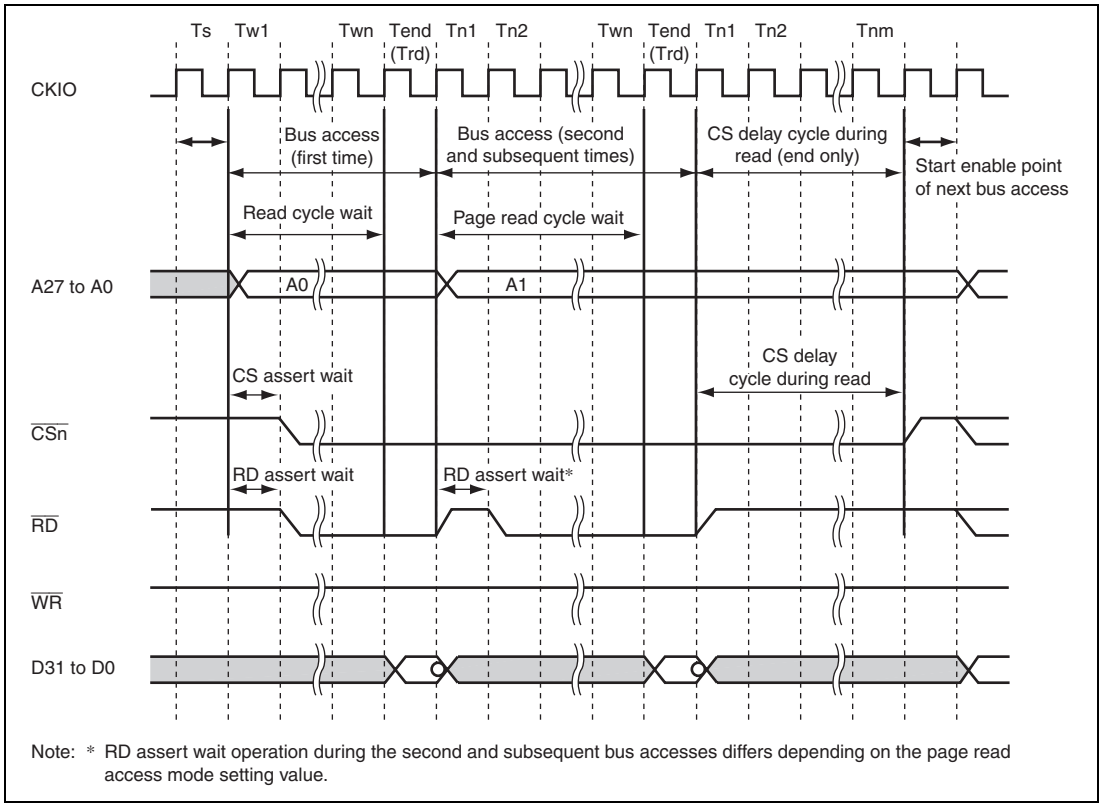


Figure 9.4 Basic Bus Timing (Page Read Operation)

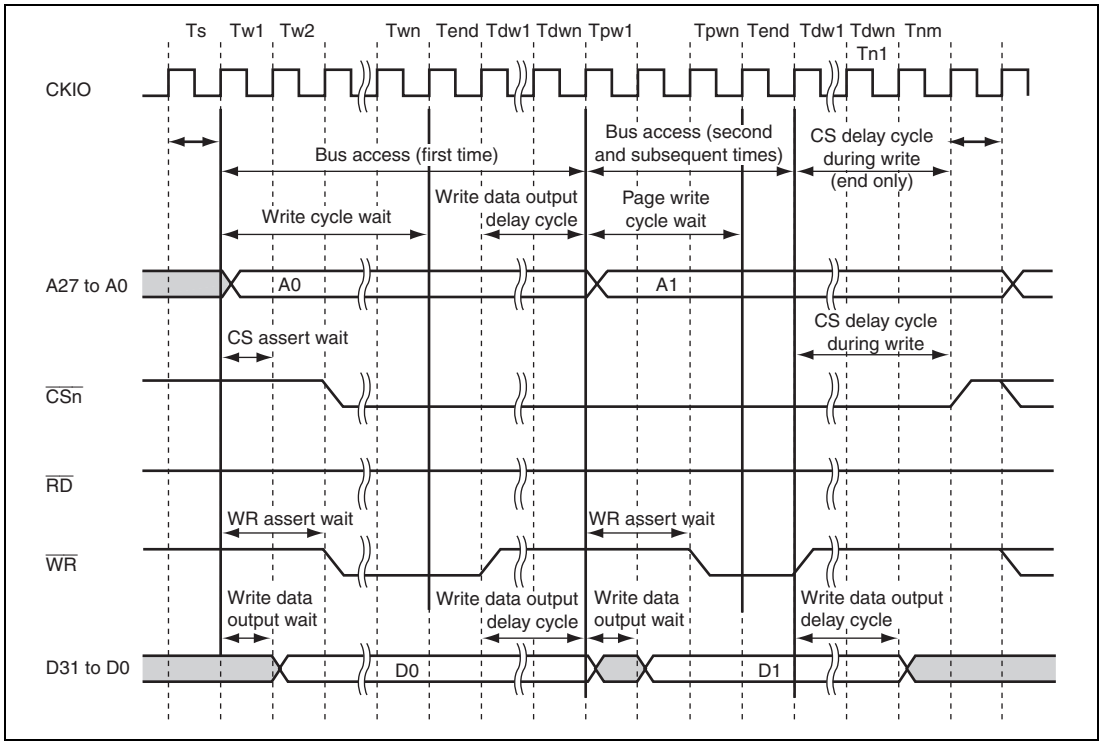


Figure 9.5 Basic Bus Timing (Page Write Operation)

1. Ts (Internal Bus Access Start)

   This is a bus access request cycle initiated by the internal bus master and with the external bus as the target. $\overline{\text{CSn}}$ is always high during this cycle. In the next cycle A27 to A0 and the write data change.

2. Tw1 to Twn (Read Cycle Wait, Write Cycle Wait)

   For the first page access, the wait operation from internal bus access start to the wait end cycle is the same as in normal access.

3. Tend (First Wait End Cycle)

   This is the final cycle in the first series of read cycle wait or write cycle wait cycles. In write access, the second and subsequent page accesses start from the next cycle, unless a write data output delay cycle has been specified (with a value other than 0). The $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal is negated (high level) in the next cycle if the RD assert wait or WD assert wait setting is other than 0. If the RD assert wait or WD assert wait setting is 0, the $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signal continues to be asserted (low level). The $\overline{\text{CSn}}$ signal is not negated and continues to be asserted (low level). In page read access, the succeeding bus access starts without waiting for the read data sample cycle (Trd).

4. Tdw1 to Tdwn (Write Data Output Delay Cycle)

   In write access write data output delay cycles are inserted between the wait end cycle and the following page access if the write data output delay wait setting is other than 0. Assertion of the address and output data is extended for the duration of this interval. Also, the $\overline{WR}$ signal is negated (high level).

5. Tpw1 to Tpwn (Page Read Cycle Wait, Page Write Cycle Wait)

   In page access the page read cycle wait and page write cycle wait settings are used in place of the read cycle wait and write cycle wait settings for the second and subsequent bus cycles. The WR assert wait setting works the same as during the first bus cycle. The RD assert wait setting operates differently depending on the page read access mode (PRMOD) setting value.

   PRMOD = 0: RD assert wait setting operates identically to first bus cycle.

   PRMOD = 1: RD assert wait setting is invalid. Operation is the same as an RD assert wait setting of 0.

6. Tend/Tdw1 to Tdwn (Wait End Cycle/Write Data Output Delay Cycle)

   These operate the same as during the first access (3 and 4 above).

7. Tn1 to Tnm (CS Delay Cycle)

   These are the cycles between the final wait end cycle and when $\overline{CSn}$ is negated (high level). The number of CS delay cycles is counted beginning from the wait end cycle.

8. Trd (Final Read Data Sample Cycle)

   This is the final sample cycle for read data.

(3) External Wait Function

The external wait signal ($\overline{WAIT}$) can be used to extend the wait cycle duration beyond the value specified by the cycle wait (CSRWAIT, CSWWAIT) or page access cycle wait (CSPRWAIT, CSPWWAIT) settings in the CSn wait control register (CSWCNTn). If external wait enable (EWENB = 1) has been selected, wait cycles are inserted for as long as the $\overline{WAIT}$ signal remains low level. The $\overline{WAIT}$ signal is disabled if external wait disable (EWENB = 0) has been selected.

Note that the wait cycles specified by the settings of the CSn wait control register (CSWCNTn) are inserted regardless of the state of the $\overline{WAIT}$ signal.

(a) Normal Read/Write Operation

The $\overline{WAIT}$ signal is sampled all the time and its result is reflected two cycles later. Thus, when the $\overline{WAIT}$ signal is low two cycles before the end of the wait cycles, external cycles are inserted. After the $\overline{WAIT}$ signal has gone high, the wait cycles end two cycles later.

(b) Page Access Operation

The initial data read/write operation is the same as a normal read/write operation. That is, when the $\overline{\text{WAIT}}$ signal is low two cycles before the end of the wait cycles (Tend), external wait cycles are inserted. After the $\overline{\text{WAIT}}$ signal has gone high, the wait cycles end (Tend) two cycles later.

In the second and subsequent read accesses, the page wait cycle is extended if the $\overline{\text{WAIT}}$ signal is low two cycles before the end of the page access wait cycle (Tend), and the page wait cycles end two cycles after the $\overline{\text{WAIT}}$ signal has gone high.

Figure 9.6 shows an example of external wait timing for page read access using longword (32-bit) access to a 16-bit channel.

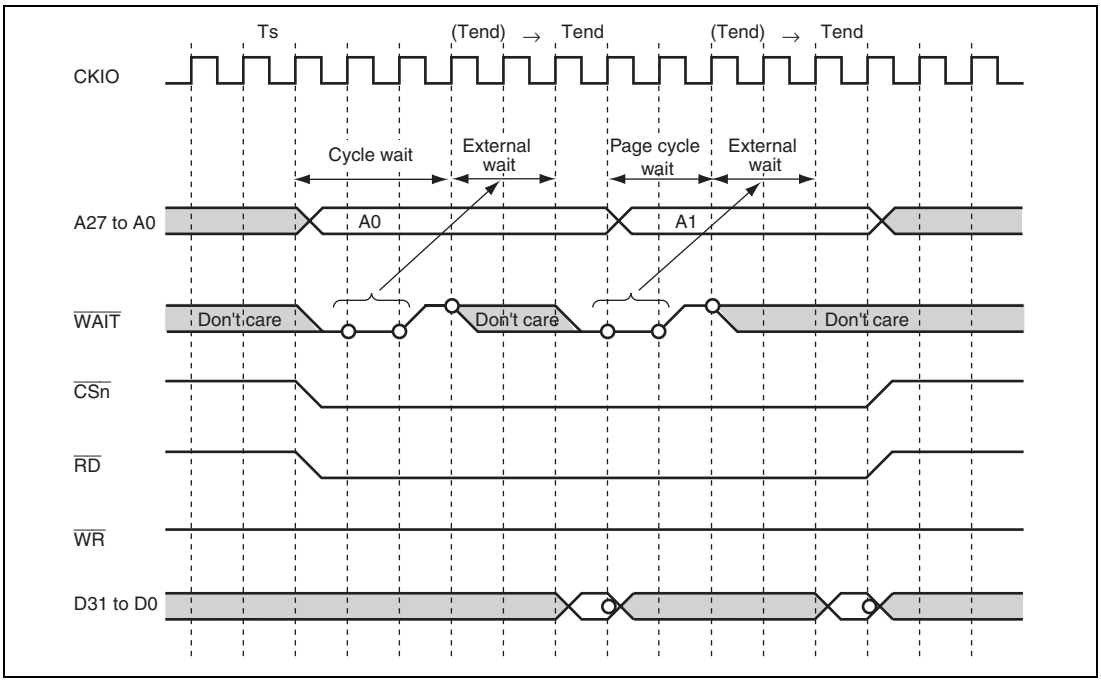


Figure 9.6 External Wait Timing Example (Page Read Access to 16-Bit Channel)

(4) Access Type and Data Alignment

(a) 32-Bit Bus Channel

If a 32-bit bus is selected by the external bus width select bits in the CSn control register, A27 to A2 are enabled as address signals for longword units and A1 and A0 are disabled (fixed low level). Table 9.7 shows the data alignment corresponding to byte addresses for different data sizes.

Pins $\overline{WR3}$ to $\overline{WR0}$ are enabled when byte strobe mode (WRMOD = 0) is selected. Pins $\overline{BC3}$ to $\overline{BC0}$ are not used.

Only the $\overline{WR3}$ pin is enabled when one-write strobe mode (WRMOD = 1) is selected. A low-level signal is output from the $\overline{WR3}$ pin during write access, regardless of the data size. At this time pins $\overline{WR2}$ to $\overline{WR0}$ are disabled (fixed high level). The valid byte positions are indicated by pins $\overline{BC3}$ to $\overline{BC0}$.

Table 9.7 Data Alignment (32-Bit Bus Channel)

		DATA				$\overline{WR}/\overline{BC}$			
Data Size	Byte Address (Lower 2 Bits)	[31:24]	[23:16]	[15:8]	[7:0]	[3]	[2]	[1]	[0]
Byte	0	O	×	×	×	L	H	H	H
	1	×	O	×	×	H	L	H	H
	2	×	×	O	×	H	H	L	H
	3	×	×	×	O	H	H	H	L
Word	0	O	O	×	×	L	L	H	H
	2	×	×	O	O	H	H	L	L
Longword	0	O	O	O	O	L	L	L	L

Note: The valid bits in the data bus for each data size are indicated by circles (O).
Crosses (×) indicate bus data bits that are undefined.

(b) 16-Bit Bus Channel

If a 16-bit bus is selected by the external bus width select bits in the CSn control register, A27 to A1 are enabled as address signals for word units and A0 is disabled (fixed low level). Table 9.8 shows the data alignment corresponding to byte addresses for different data sizes.

Pins $\overline{WR1}$ and $\overline{WR0}$ are enabled when byte strobe mode (WRMOD = 0) is selected. Pins $\overline{WR3}$ and $\overline{WR2}$ are disabled. Pins $\overline{BC3}$ to $\overline{BC0}$ are not used.

Only the $\overline{WR1}$ pin is enabled when one-write strobe mode (WRMOD = 1) is selected. A low-level signal is output from the $\overline{WR1}$ pin during write access, regardless of the data size. At this time the $\overline{WR0}$ pin is disabled (fixed high level). The valid byte positions are indicated by pins $\overline{BC1}$ and $\overline{BC0}$.

Table 9.8 Data Alignment (16-Bit Bus Channel)

		DATA				$\overline{WR}/\overline{BC}$			
Data Size	Byte Address (Lower 2 Bits)	[31:24]	[23:16]	[15:8]	[7:0]	[3]	[2]	[1]	[0]
Byte	0	×	×	O	×	*	*	L	H
	1	×	×	×	O	*	*	H	L
	2	×	×	O	×	*	*	L	H
	3	×	×	×	O	*	*	H	L
Word	0	×	×	O	O	*	*	L	L
	2	×	×	O	O	*	*	L	L
Longword	0 (1st)	×	×	O	O	*	*	L	L
	2 (2nd)	×	×	O	O	*	*	L	L

Note: The valid bits in the data bus for each data size are indicated by circles (O).
Crosses (×) indicate bus data bits that are undefined.
Asterisks (*) indicate write/byte control bits that are disabled (fixed high level).

(c) 8-Bit Bus Channel

If an 8-bit bus is selected by the external bus width select bits in the CSn control register, A27 to A0 are enabled as address signals for byte units. Table 9.9 shows the data alignment corresponding to byte addresses for different data sizes.

With an 8-bit bus channel only the $\overline{WR0}$ pin is enabled, regardless of the strobe mode setting. A low-level signal is output to $\overline{WR0}$ during write access. $\overline{BC0}$ constantly outputs low level. Pins $\overline{WR3}$ to $\overline{WR1}$ and pins $\overline{BC3}$ to $\overline{BC1}$ are not used.

Table 9.9 Data Alignment (8-Bit Bus Channel)

		DATA				$\overline{WR}/\overline{BC}$			
Data Size	Byte Address (Lower 2 Bits)	[31:24]	[23:16]	[15:8]	[7:0]	[3]	[2]	[1]	[0]
Byte	0	×	×	×	O	*	*	*	L
	1	×	×	×	O	*	*	*	L
	2	×	×	×	O	*	*	*	L
	3	×	×	×	O	*	*	*	L
Word	0 (1st)	×	×	×	O	*	*	*	L
	1 (2nd)	×	×	×	O	*	*	*	L
	2 (1st)	×	×	×	O	*	*	*	L
	3 (2nd)	×	×	×	O	*	*	*	L
Longword	0 (1st)	×	×	×	O	*	*	*	L
	1 (2nd)	×	×	×	O	*	*	*	L
	2 (3rd)	×	×	×	O	*	*	*	L
	3 (4th)	×	×	×	O	*	*	*	L

Note: The valid bits in the data bus for each data size are indicated by circles (O).
Crosses (×) indicate bus data bits that are undefined.
Asterisks (*) indicate write/byte control bits that are disabled (fixed high level).

9.5.2 SDRAM Interface

A description is provided here of the SDRAM controller (SDRAMC) operation enable and SDRAM bus width settings as well as operations involving SDRAM (read, write, auto-refresh, self-refresh, initialization sequence, and mode register settings).

(1) SDRAM Access Enable/Disable and SDRAM Bus Width Settings

Enabling and disabling SDRAM access is performed by making settings in the individual SDRAMCm control registers to enable or prohibit SDRAMC operation. SDRAM bus width settings are also performed by means of the SDRAMCm control registers.

Even if the SDRAMC control register is set to disable SDRAMC operation, refresh operation will still take place if self-refresh or auto-refresh operation is set as enabled.

(2) SDRAM Commands

SDRAMC controls the SDRAM by issuing commands each bus cycle. These commands are defined by combinations of RAS, CAS, WE, CKE, CS, etc.

Table 9.10 lists the commands issued by SDRAMC.

Table 9.10 SDRAMC Commands

	Command	SDCS	SDRAS	SDCAS	SDWE	SDCKE	BA1	BA0
DSL	Deselect	H	X	X	X	X	X	X
ACT	Initialize row and bank	L	L	H	H	H	V	V
RD	Read	L	H	L	H	H	V	V
WR	Write	L	H	L	L	H	V	V
PRA	Precharge all banks	L	L	H	L	H	X	X
RFA	Auto-refresh	L	L	L	H	H	X	X
MRS	Mode register set	L	L	L	L	H	L	L
EMRS	Extended mode register set	L	L	L	L	H	H	L
RFS	Self-refresh entry	L	L	L	H	H → L	X	X
RFX	Self-refresh exit	H	X	X	X	L → H	X	X
DPD	Deep-power-down	L	H	H	L	H → L	X	X
DPDX	Deep-power-down exit	X	X	X	X	L → H	X	X

[Legend]
H: High level, L: Low level, V: Valid, X: Don't care

(3) SDRAMC Register Setting Conditions

Rewriting of SDRAMC registers should only be performed when all of the conditions listed in table 9.11 are satisfied.

Table 9.11 Register Rewrite Conditions

Function/Operation	Register	Conditions
Self-refresh	SDRFCNT0	• SDRAM access disabled (set in SDRAMCm*[1]) • Auto-refresh enabled (DRFEN = 1) • Power-down disabled (DPWD/DPWDCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Auto-refresh	SDRFCNT1	• Self-refresh disabled (DSFEN/DSFENCI = 0) • Power-down disabled (DPWD/DPWDCI = 0)
Initialization sequence	SDIR0	• Before start of initialization sequence
	SDIR1	• After reset or after recovery from deep-power-down
Power-down	SDPWDCNT	• SDRAM access disabled (set in SDRAMCm*[1]) • Auto-refresh enabled (DRFEN = 1) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Deep-power-down	SDDPDCNT	• SDRAM access disabled (set in SDRAMCm*[1]) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Auto-refresh disabled (DRFEN = 0) • Power-down disabled (DPWD/DPWDCI = 0)
Address register settings	SD0ADR, SD1ADR	• Auto-refresh disabled (DRFEN = 0) • SDRAM access disabled (set in SDRAMCm*[1]) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Power-down disabled (DPWD/DPWDCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Timing register settings	SD0TR, SD1TR	• Self-refresh in progress (DSFEN/DSFENCI = 1) or • Self-refresh disabled (DSFEN/DSFENCI = 0) • Auto-refresh disabled (DRFEN = 0) • SDRAM access disabled (set in SDRAMCm*[1])

Function/Operation	Register	Conditions
Mode register settings	SD0MOD, SD1MOD*2	• SDRAM access disabled (set in SDRAMCm*1) • Self-refresh disabled (DSFEN/DSFENCI = 0) • Power-down disabled (DPWD/DPWDCI = 0) • Deep-power-down disabled (DDPD/DDPDCI = 0)
Clock stop control signal settings	SDCKSCNT	• Deep-power-down disabled (DDPD/DDPDCI = 0)

Notes: 1. After writing 0 to EXENB, check to confirm that the EXENB bit has been cleared to 0.
2. Do not fail to confirm that all status bits in the SDRAM status register (SDSTR) have been cleared to 0 before rewriting this bit.

(4) Self-Refresh

Transition to and from self-refresh mode is controlled by means of settings to SDRAM refresh control register 0 (SDRFCNT0). Transition to and from self-refresh mode takes place simultaneously for all channels.

An auto-refresh cycle operation takes place immediately before transition to self-refresh mode. While in self-refresh mode the CKE signal is low level. Immediately after recovery from self-refresh mode an auto-refresh cycle is triggered.

Figure 9.7 shows the timing of transition to self-refresh mode, and figure 9.8 shows the timing of recovery from self-refresh mode.

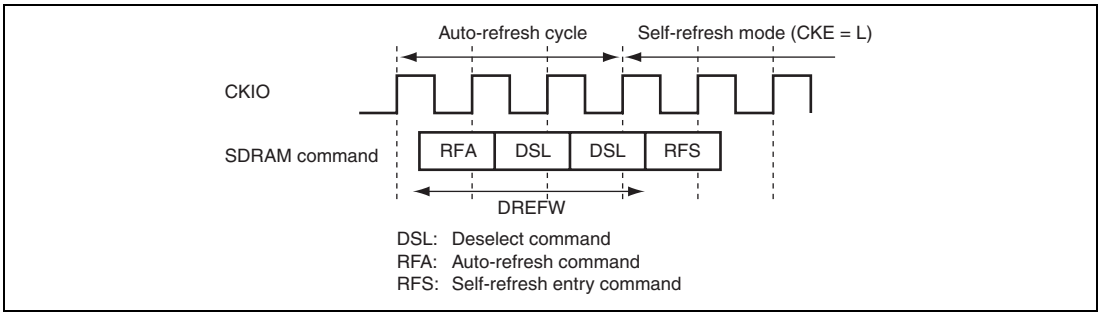


Figure 9.7 Example of Timing of Transition to Self-Refresh Mode (DREFW Bit Set Value: 0010)

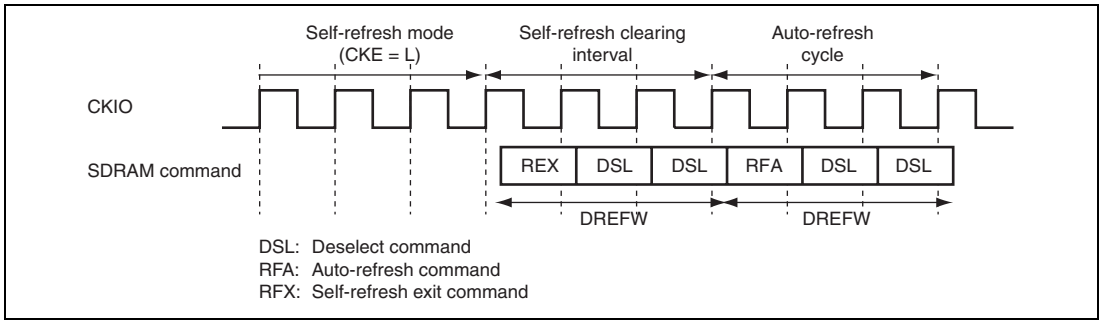


Figure 9.8 Example of Timing of Recovery from Self-Refresh Mode (DREFW Bit Set Value: 0010)

(5) Auto-Refresh

An auto-refresh cycle starts when the auto-refresh operation enable bit (DRFEN) in SDRAM refresh control register 1 (SDRFCNT1) is set to 1. After that refresh requests are issued at fixed intervals, activating auto-refresh cycles. However, the activation of auto-refresh cycles may sometimes be delayed because refresh requests are not accepted during read or write accesses.

A refresh request is issued immediately if the auto-refresh operation enable bit (DRFEN) in SDRAM refresh control register 1 (SDRFCNT1) is set to 1 while auto-refresh is enabled.

The refresh counter is halted in self-refresh or deep-power-down mode. After recovery from self-refresh or deep-power-down mode an auto-refresh cycle is activated, after which the counter value is reset and the counter begins operating again

Make auto-refresh settings in SDRAM refresh control register 1 (SDRFCNT1). Note that refresh cycles affect all SDRAM channels. Figure 9.9 shows an auto-refresh cycle timing example.

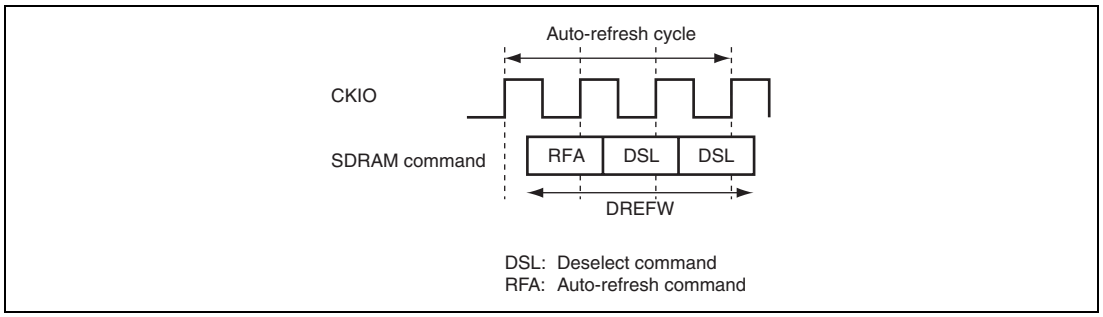


Figure 9.9 Auto-Refresh Cycle Timing Example (DREFW Bit Set Value: 0010)

(6) Initialization Sequencer

SDRAMC is provided with a sequencer for issuing the commands for SDRAM initialization. The initialization sequence should always be initiated a single time only following a reset (all channels) and following recovery from deep-power-down mode (individual channels). In such cases operation cannot be guaranteed if the initialization sequence is not performed, or if it is performed more than once.

The SDRAM initialization sequence issues the precharge-all-banks command followed by n (n = 1 to 15) auto-refresh commands, in that order. Make timing settings for the initialization sequencer to SDRAM initialization register 0 (SDIR0). Initialization sequences are initiated using SDRAM initialization register 1 (SDIR1).

Note that an initialization sequence for all channels is initiated using the DINIRQ bit.

Figure 9.10 shows a timing example for the initialization sequence. Setting DARFC to specify two or more times causes multiple initialization auto-refresh cycles to be performed.

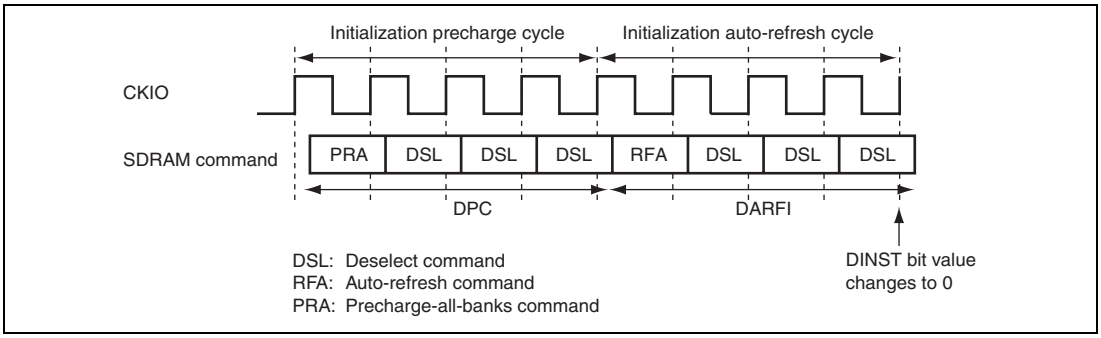


Figure 9.10 Initialization Sequence Timing Example (DPC Bit Set Value: 001, DARFI Bit Set Value: 0001, DARFC Bit Set Value: 001)

(7) Power-Down Mode

SDRAMC supports an SDRAM power-down mode. In power-down mode the SDCKE signal from SDRAMC goes low level. While in power-down mode auto-refresh operations are performed at the interval specified by the auto-refresh request interval setting (DRFC) bits in SDRAM refresh control register 1 (SDRFCNT1). The SDCKE signal only goes high when an auto-refresh command is issued.

Transition to and recovery from power-down mode are performed using the SDRAM power-down control register (SDPWDCNT).

Setting the DPWD bit to 1 causes SDRAMC to transition to power-down mode. Clearing the DPWD bit to 0 causes SDRAMC to recover from power-down mode.

The SDCKE signal from SDRAMC goes high level when recovery from power-down mode occurs.

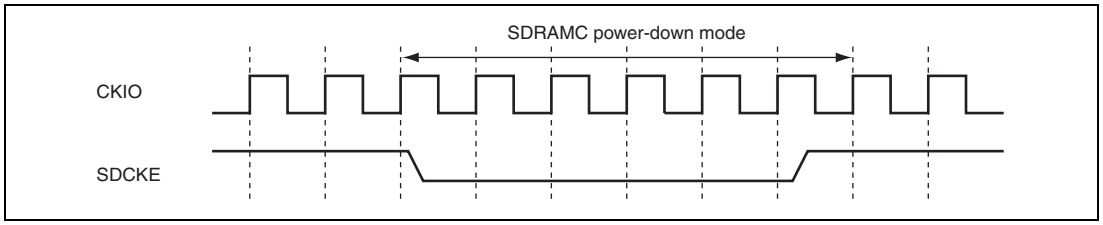


Figure 9.11 SDRAMC Power-Down Mode

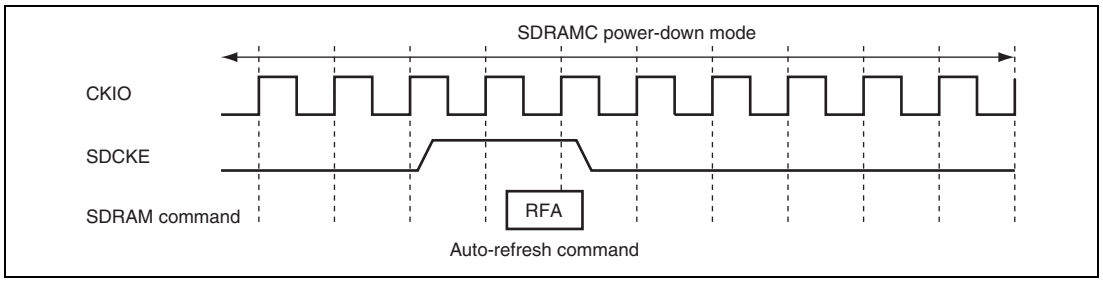


Figure 9.12 Auto-Refresh Operation in SDRAMC Power-Down Mode

(8) Deep-Power-Down Mode

SDRAMC supports an SDRAM deep-power-down mode. In deep-power-down mode SDRAMC issues a deep-power-down command and drives the SDCKE signal low level.

Transition to and recovery from deep-power-down mode are performed using the SDRAM deep-power-down control register (SDDPDCNT).

Setting the DDPD bit to 1 causes SDRAMC to put all channels into deep-power-down mode. Clearing the DDPD bit to 0 causes SDRAMC to recover from deep-power-down mode.

During recovery from deep-power-down mode, SDRAMC issues a deep-power-down exit command and drives the SDCKE signal high level.

Following recovery from deep-power-down exit, wait for the duration designated for the SDRAM being used and then execute an initialization sequence.

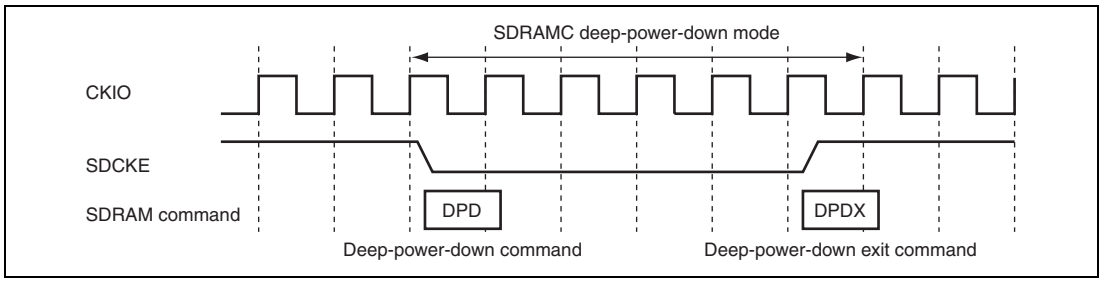


Figure 9.13 SDRAMC Deep-Power-Down Mode

(9) Read/Write Access

The following two types of read/write access are supported.

- Multiple read/multiple write
- Single read/single write

Multiple read/multiple write occurs in the following cases.

1. CPU burst access (cache replace)
2. Access with longword (32-bit) to the SDRAM data bus having 8-bit or 16-bit width
3. Access with word (16-bit) to the SDRAM data bus having 8-bit width
4. Multiple data transfer in DMA pipeline transfer

The access timing can be set independently for each channel using the SDRAMI timing register (SDITR). Access timing examples are described below.

(a) Multiple Read/Multiple Write Access

Figure 9.14 shows a timing example for multiple read of 4 units of data, and figure 9.15 for multiple write of 4 units of data.

The number of DMA transfers performed will vary depending on factors such as the number of transfers and the transfer data size per operand and the SDRAM bus width. Read commands or write commands may or may not be issued consecutively in response to an access request from the bus master. When read commands or write commands are not issued consecutively, a deselect command is issued between them.

Furthermore, deactivation and activation are performed automatically when the SDRAM row address changes during a DMA transfer operation.

Figure 9.16 shows a timing example for multiple read of 4 units of data, and figure 9.17 for multiple write of 4 units of data, when read/write commands are not issued consecutively. Figure 9.18 shows a timing example for multiple write with a row address change.

The access timing is modified by means of settings in the SDRAMm timing register (SDmTR).

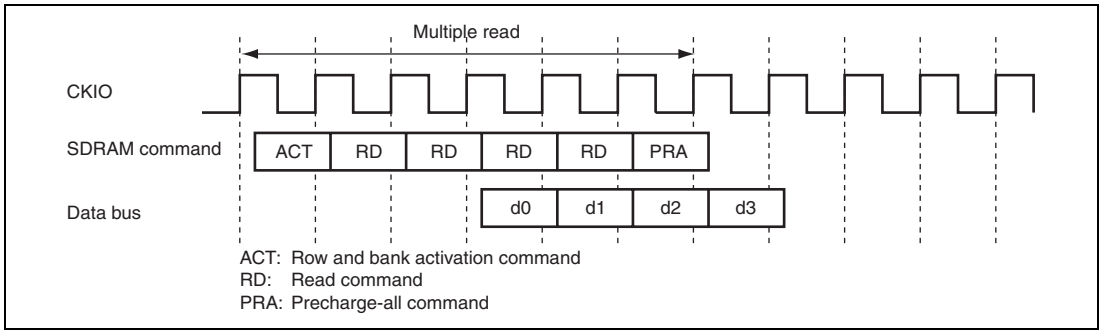


Figure 9.14 Multiple Read Timing Example (Multiple Read of 4 Data Units, Shortest Timing Settings) Consecutive Read Commands Issued

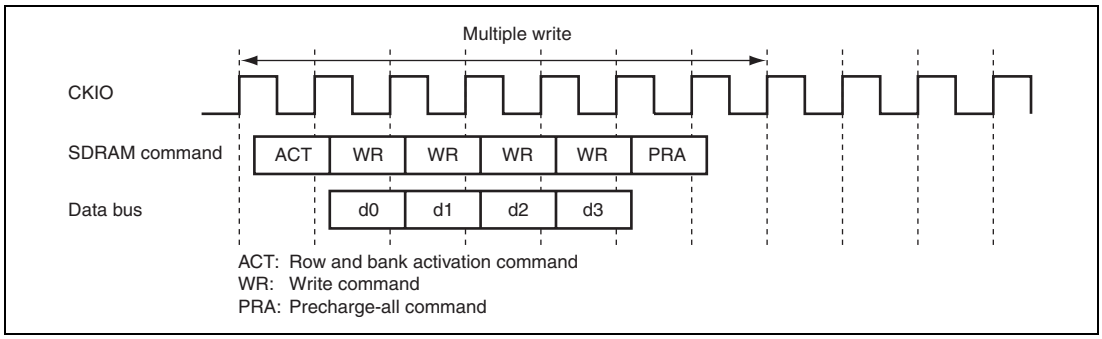


Figure 9.15 Multiple Write Timing Example (Multiple Write of 4 Data Units, Shortest Timing Settings) Consecutive Write Commands Issued

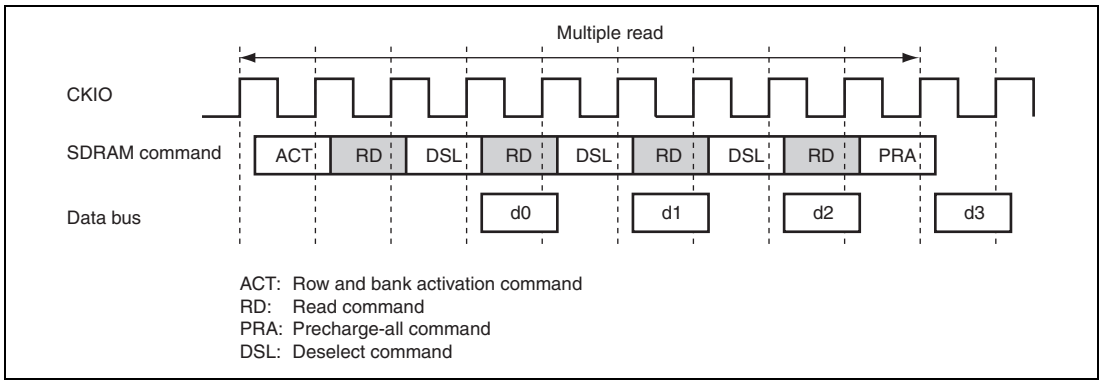


Figure 9.16 Multiple Read Timing Example (Multiple Read of 4 Data Units, Shortest Timing Settings) Non-Consecutive Read Commands Issued

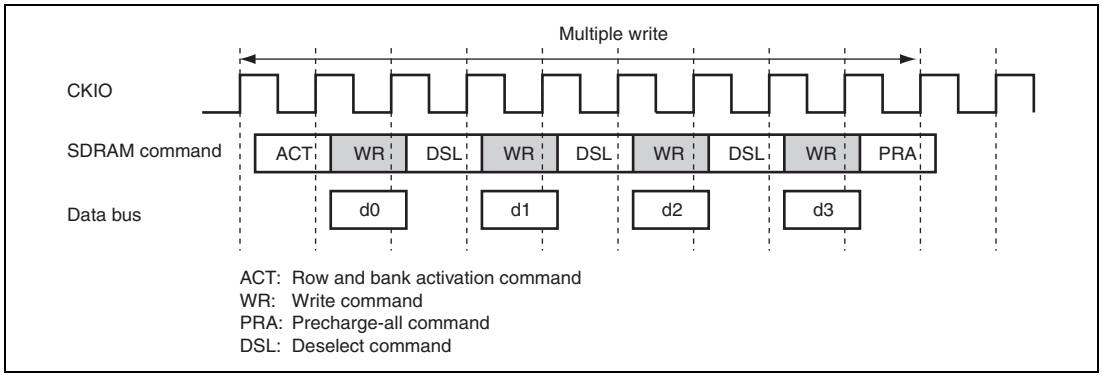


Figure 9.17 Multiple Write Timing Example (Multiple Write of 4 Data Units, Shortest Timing Settings) Non-Consecutive Write Commands Issued

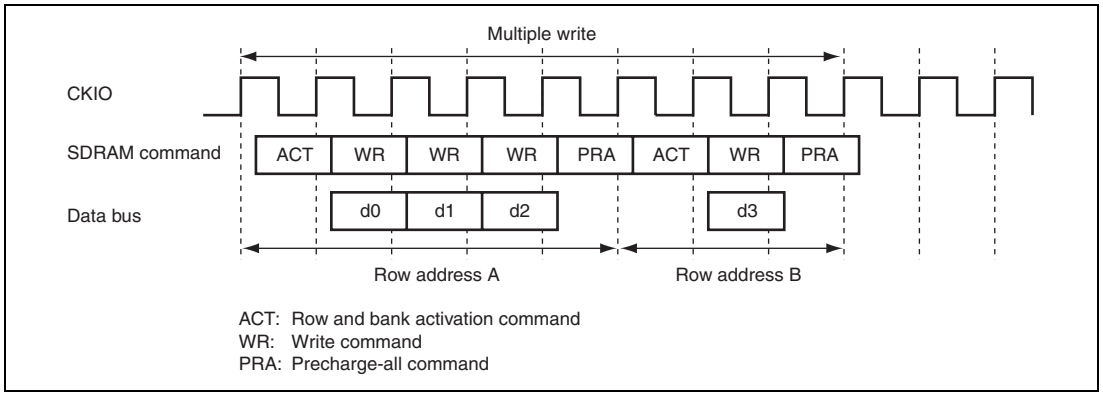


Figure 9.18 Multiple Write Timing Example (Multiple Write of 4 Data Units, Shortest Timing Settings) Access Spanning Rows

(b) Single Read/Single Write Access

Figure 9.19 shows a timing example for single read operation and figure 9.20 for single write operation. The access timing is modified by means of settings in the SDRAMm timing register (SDmTR).

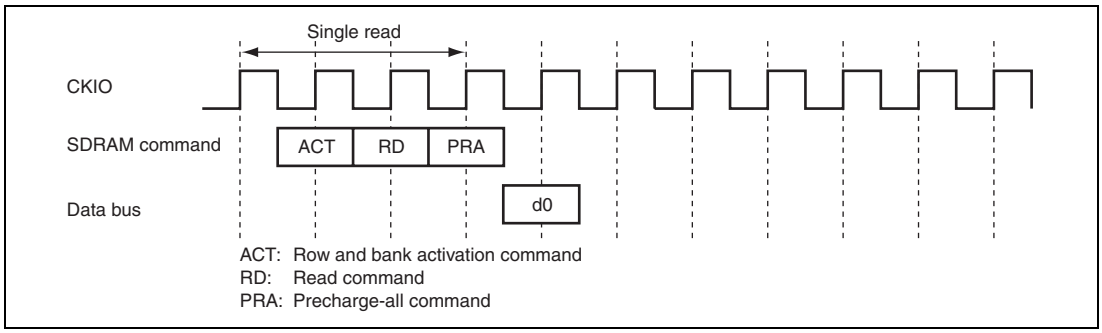


Figure 9.19 Single Read Timing Example (Shortest Timing Settings)

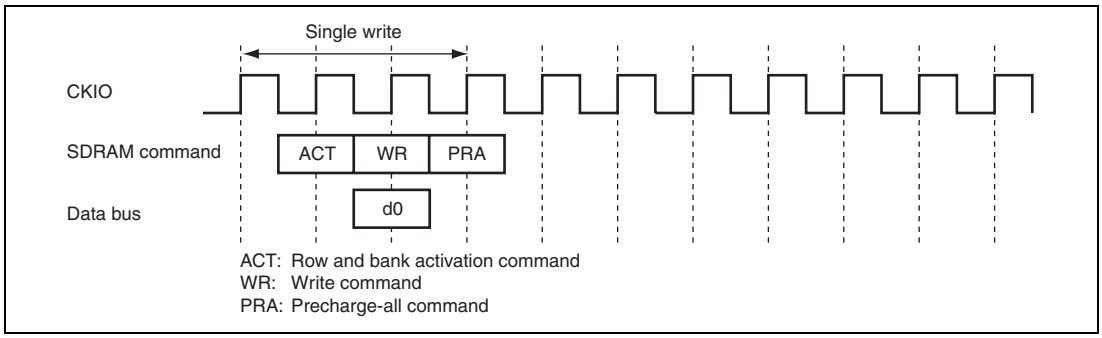


Figure 9.20 Single Write Timing Example (Shortest Timing Settings)

(c) Byte Access Control by DQM

Figures 9.21 and 9.22 show timing examples for byte accesses to the SDRAM with a 16-bit bus width. In the SDRAM access, the DQM signal is asserted when data is masked.

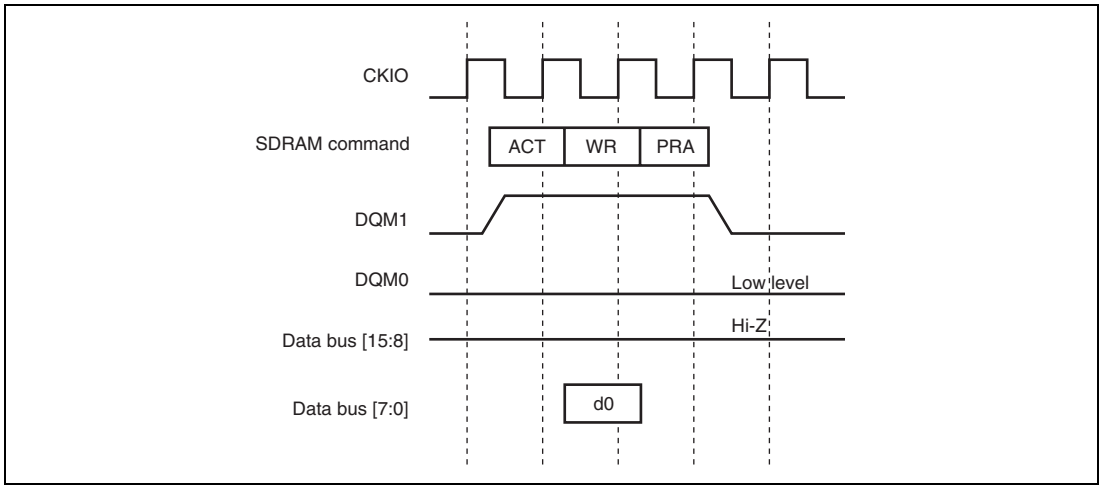


Figure 9.21 Byte Write Timing to SDRAM with 16-Bit Bus Width (Example)

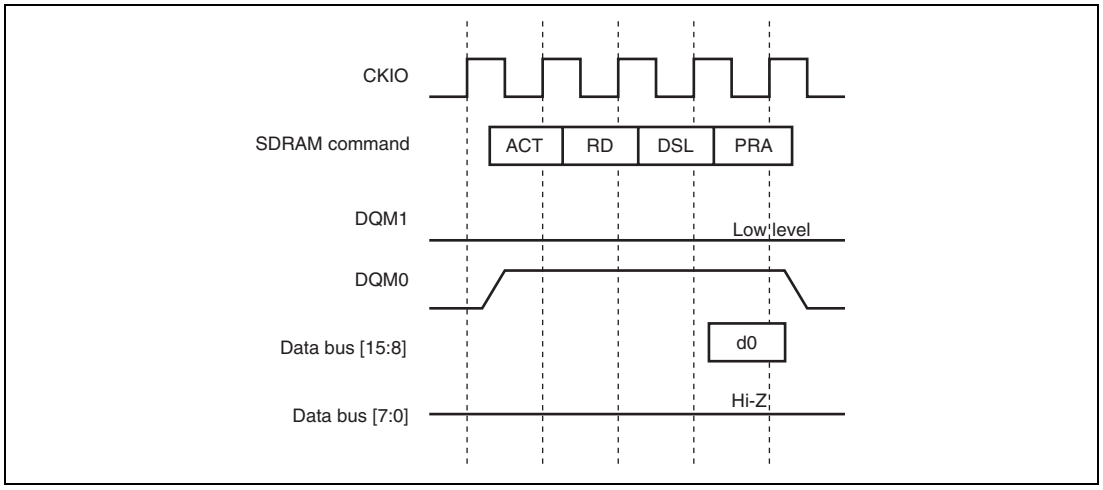


Figure 9.22 Byte Read Timing from SDRAM with 16-Bit Bus Width (Example)

(10) Mode Register Setting

Writing to the SDRAMm mode register (SDmMOD) causes mode register set commands and extended mode register set commands to be issued to the various channels. Settings to the SDRAMm mode register (SDmMOD) should be made individually for each channel.

Figure 9.23 shows the operation timing for mode register setting.

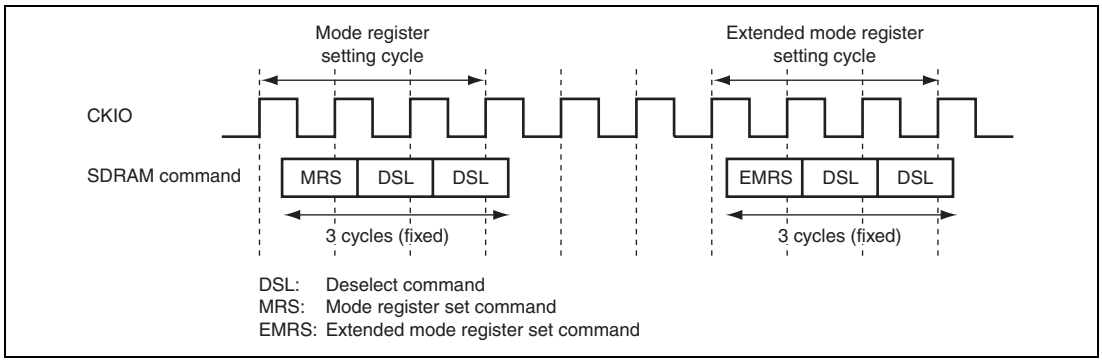


Figure 9.23 Operation Timing for Mode Register Setting

(11) Clock Stop Control Signal

SDRAMC outputs a clock stop control signal (CLKSTOP). CLKSTOP can be enabled or disabled using the DCKSEN bit in the SDRAM clock stop control signal setting register (SDCKSCNT).

The CLKSTOP signal remains low level when the clock stop control signal is disabled.

When clock stop control signal is enabled, the CLKSTOP and CKIO signals operate in conjunction with transition to and recovery from deep-power-down mode.

During a transition to deep-power-down mode, the CLKSTOP signal goes high after the deep-power-down entry command is issued. During a recovery from deep-power-down mode, the CLKSTOP signal goes low and a deep-power-down exit command is issued when the clearing of the DDPD bit to 0 is accepted by SDRAMC and the CKIO starts operation.

DCKSC, the period between the change of CLKSTOP along with CKIO and the issuance of deep power-down entry or exit command, can be set by the SDRAM clock stop control signal setting register.

Figures 9.24 and 9.25 show the operation timing of the clock stop control signal.

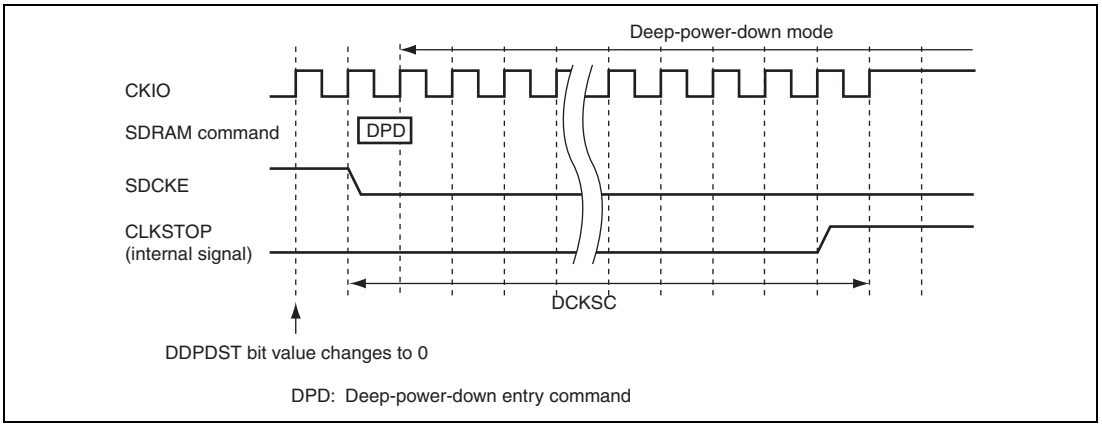


Figure 9.24 Clock Stop Control Signal Operation Timing (Transition to Deep-Power-Down Mode)

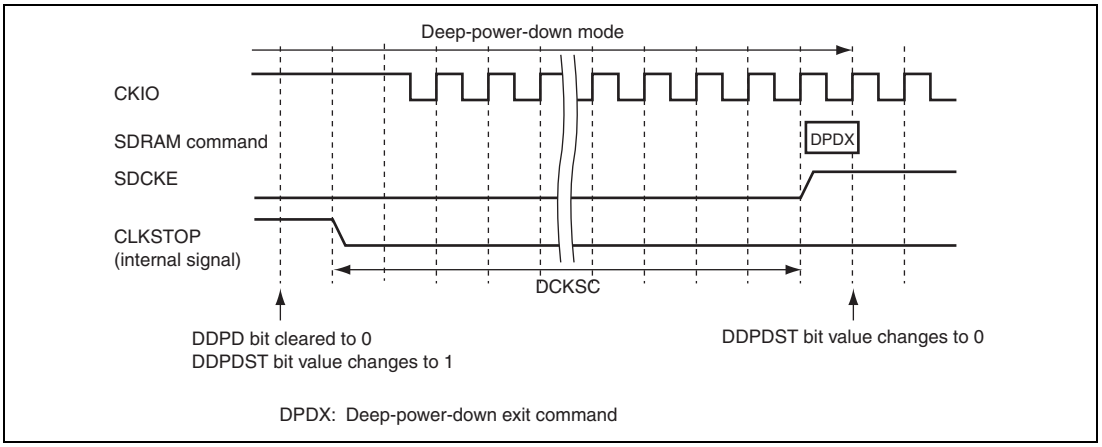


Figure 9.25 Clock Stop Control Signal Operation Timing (Recovery from Deep-Power-Down Mode)

(12) SDRAMC Setting Examples

The SDRAMC setting procedure, timing register setting examples, and the procedure for transitioning to and recovering from self-refresh mode, power-down mode, and deep-power-down mode are described below.

(a) SDRAMC Setting Procedure

Figure 9.26 shows the SDRAMC setting procedure.

Note that the specifications of the power-up sequence, etc., may vary depending on the SDRAM used. Study the SDRAM specifications carefully before making system settings.

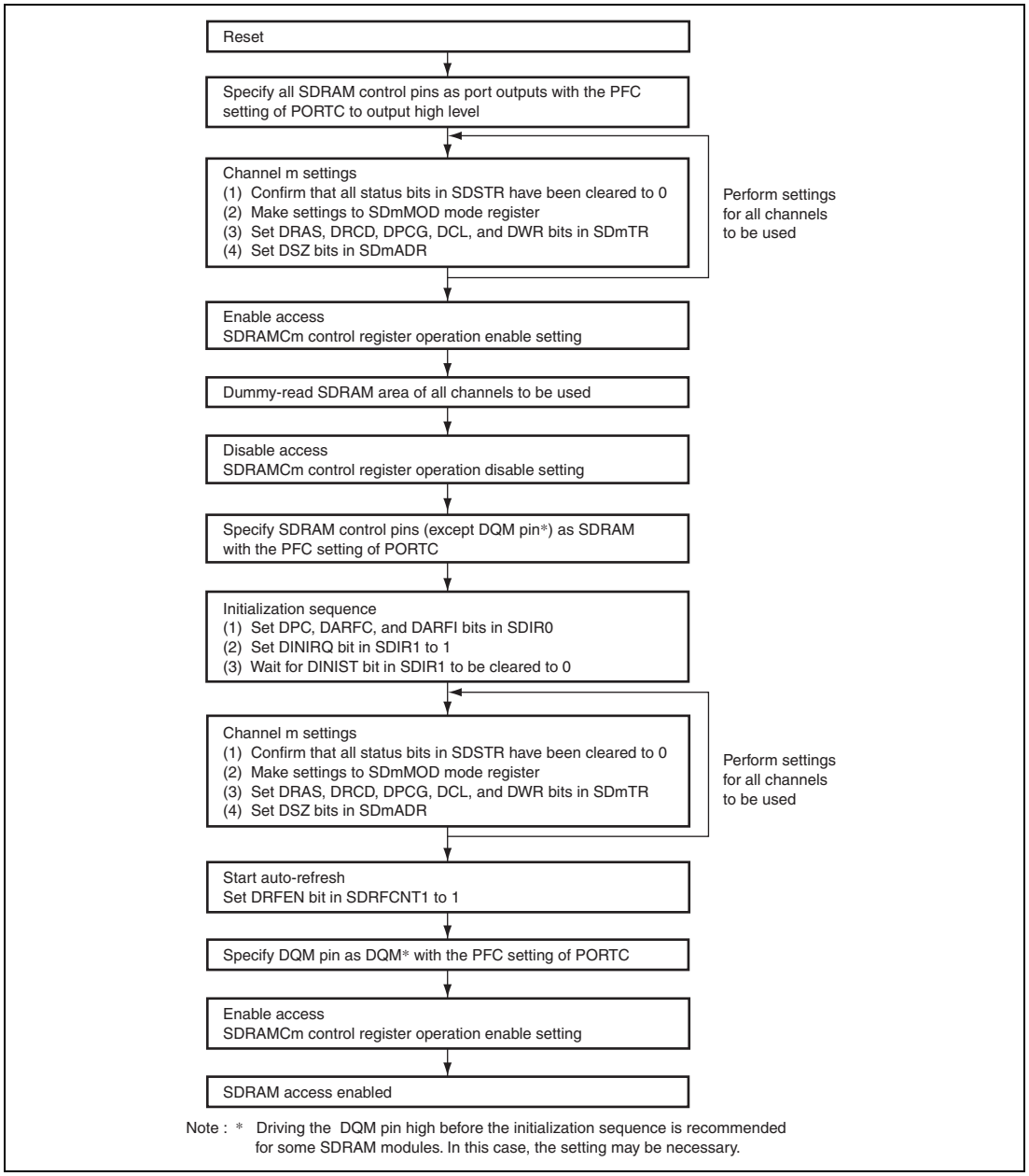


Figure 9.26 SDRAMC Setting Procedure

(b) Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 9.27 shows the procedure for transitioning to and recovering from self-refresh mode.

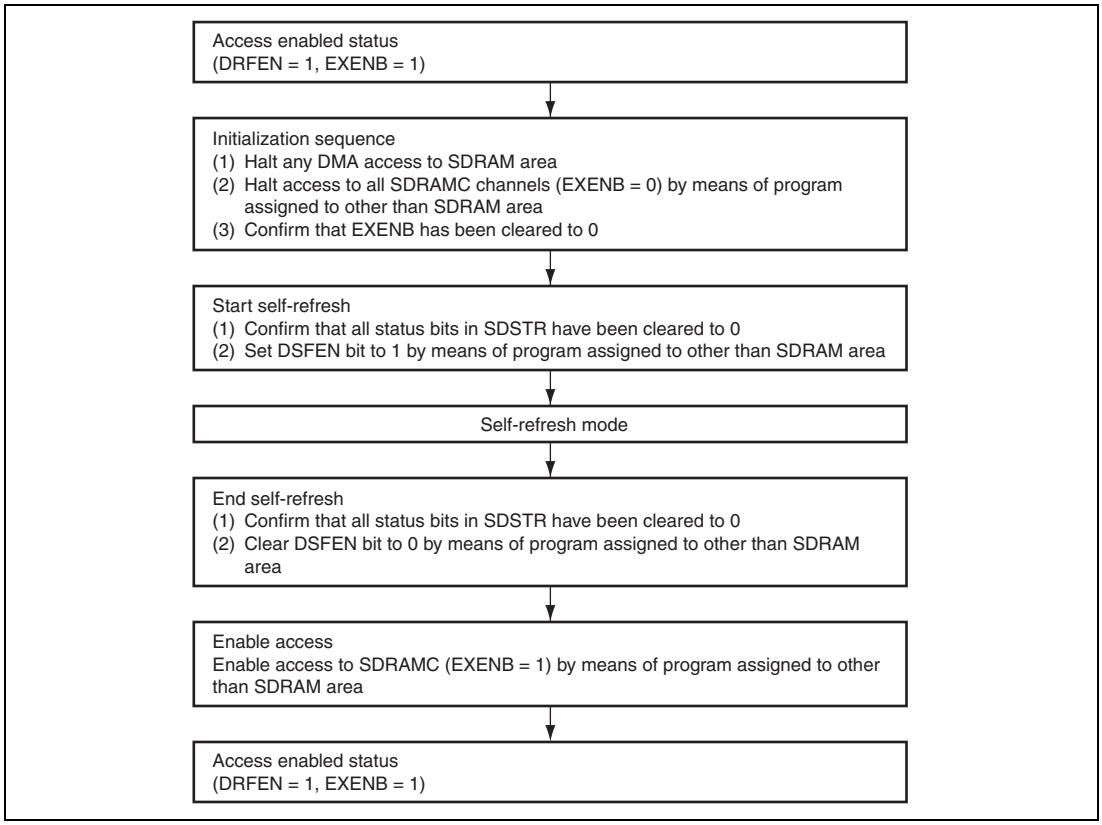


Figure 9.27 Procedure for Transition to and Recovery from Self-Refresh Mode

Note: Before transitioning to or recovering from self-refresh mode it is necessary to halt SDRAM access to the affected channels. Consequently, it is not possible to transition to or recover from self-refresh mode while programs or DMA operations that access SDRAM are in progress. Pay attention to the following points when writing programs.

- Before transitioning to self-refresh mode, halt any DMA channel transfers that access the SDRAM area of the affected channels.
- Make sure that programs run while transitioning to self-refresh mode, while in self-refresh mode, or while recovering from self-refresh mode do not access operands or fetch (or pre-fetch) instructions stored in the SDRAM area.

(c) Procedure for Transition to and Recovery from Deep-Power-Down Mode

Figure 9.28 shows the procedure for transitioning to deep-power-down mode.

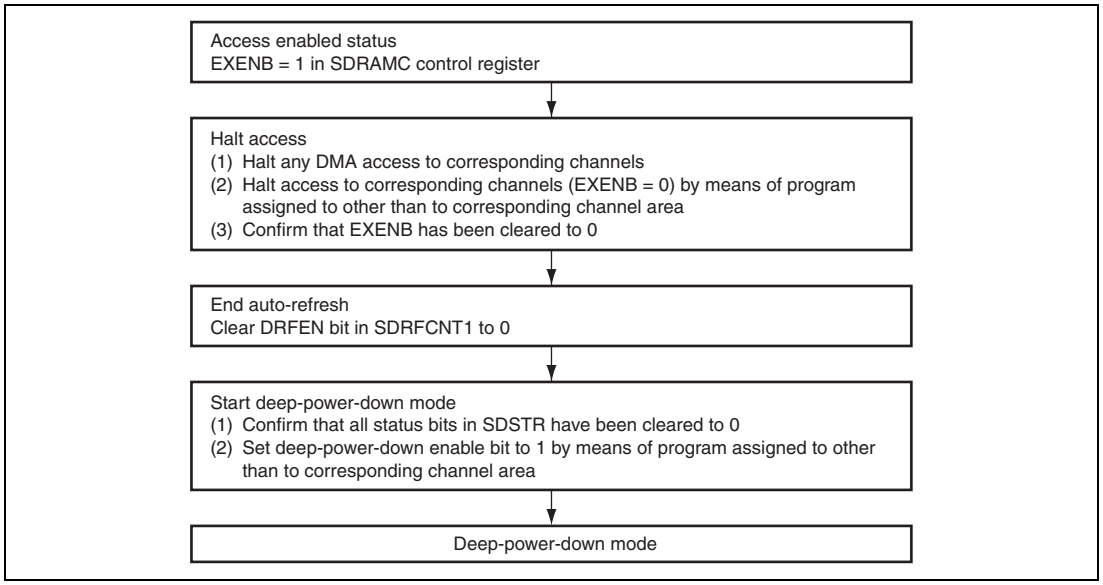


Figure 9.28 Procedure for Transition to Deep-Power-Down Mode

Figure 9.29 shows the procedure for recovering from deep-power-down mode.

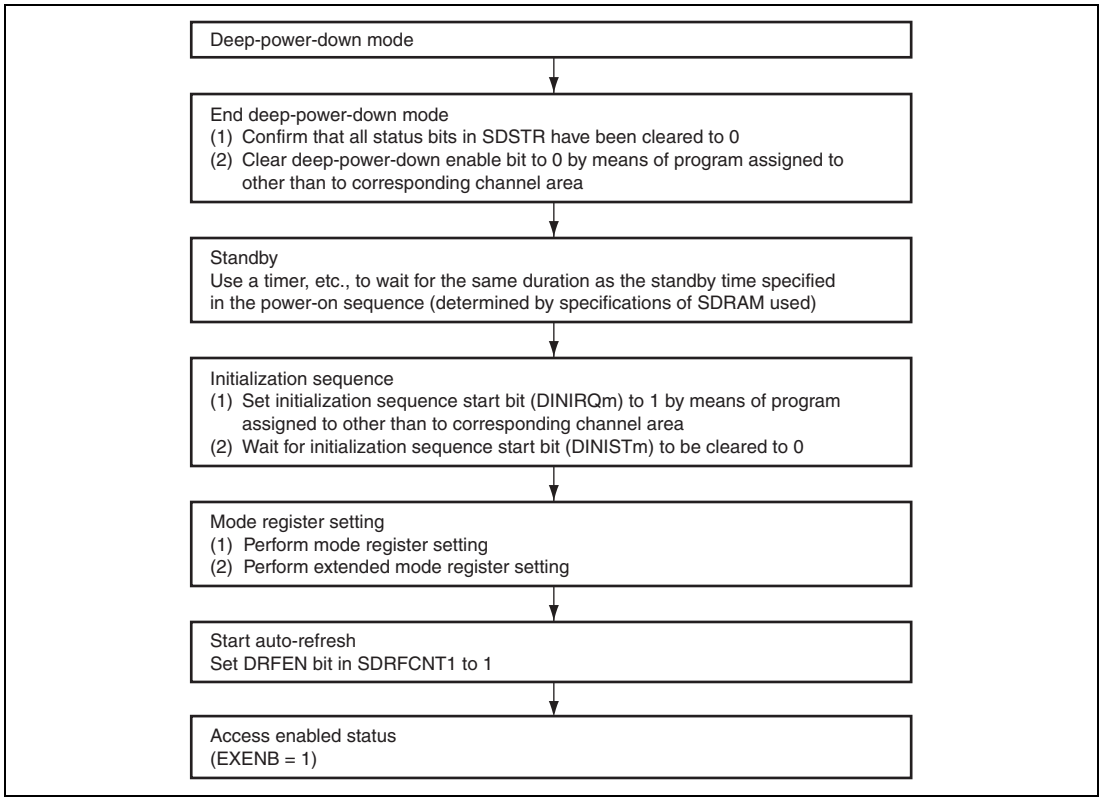


Figure 9.29 Procedure for Recovery from Deep-Power-Down Mode

Note: Before transitioning to or recovering from deep-power-down mode it is necessary to halt SDRAM access to the affected channels. Consequently, it is not possible to transition to or recover from deep-power-down mode while programs or DMA operations that access SDRAM are in progress. Pay attention to the following points when writing programs.

- Before transitioning to deep-power-down mode, halt any DMA channel transfers that access the SDRAM area of the affected channels.
- Make sure that programs run while transitioning to deep-power-down mode, while in deep-power-down mode, or while recovering from deep-power-down mode do not access operands or fetch (or pre-fetch) instructions stored in the SDRAM area.

(d) Timing Register Set Values and Access Timing

The correspondence between the SDRAMm timing register (SDmTR) set values and the read and write access timing is described below.

- Multiple Read Timing Setting Examples

  Figures 9.30 to 9.32 show the correspondence between the timing of multiple read operations involving 4 data units and the set values of the SDRAMm timing register (SDmTR). Table 9.12 shows the SDRAMm timing register (SDmTR) set values for each figure.

Table 9.12 SDITR Set Value Correspondence Table (Multiple Read Timing)

Figure	DRAS	DRCD	DPCG	DCL
Figure 9.30	010	00	001	010
Figure 9.31	000	01	001	010
Figure 9.32	000	01	001	011

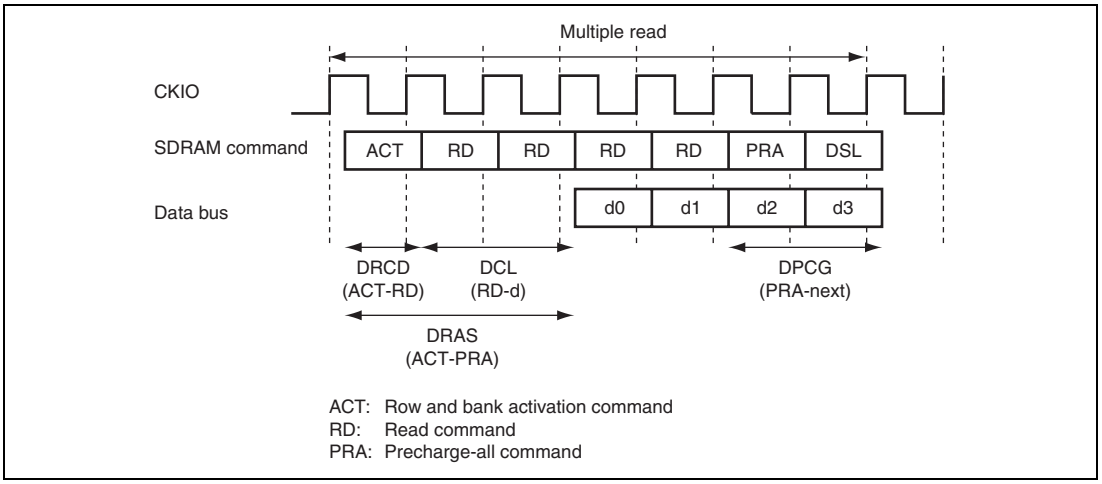


Figure 9.30 Multiple Read Timing Example 1

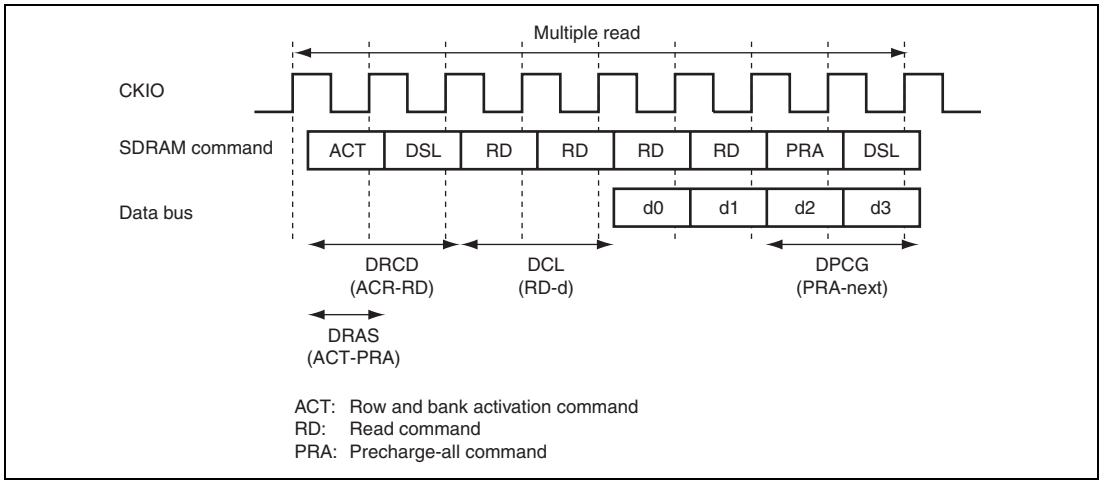


Figure 9.31 Multiple Read Timing Example 2

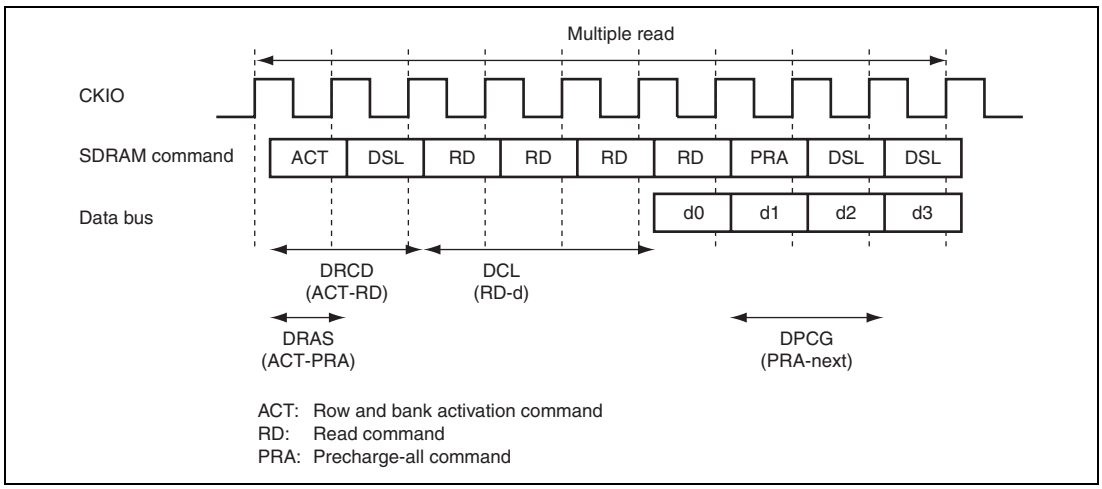


Figure 9.32 Multiple Read Timing Example 3

- Multiple Write Timing Setting Examples

  Figures 9.33 to 9.35 show the correspondence between the timing of multiple write operations involving 4 data units and the set values of the SDRAMm timing register (SDmTR). Table 9.13 shows the SDRAMm timing register (SDmTR) set values for each figure.

Table 9.13 SDITR Set Value Correspondence Table (Multiple Write Timing)

Figure	DRAS	DRCD	DPCG	DWR
Figure 9.33	010	00	001	0
Figure 9.34	000	01	001	0
Figure 9.35	000	01	001	1

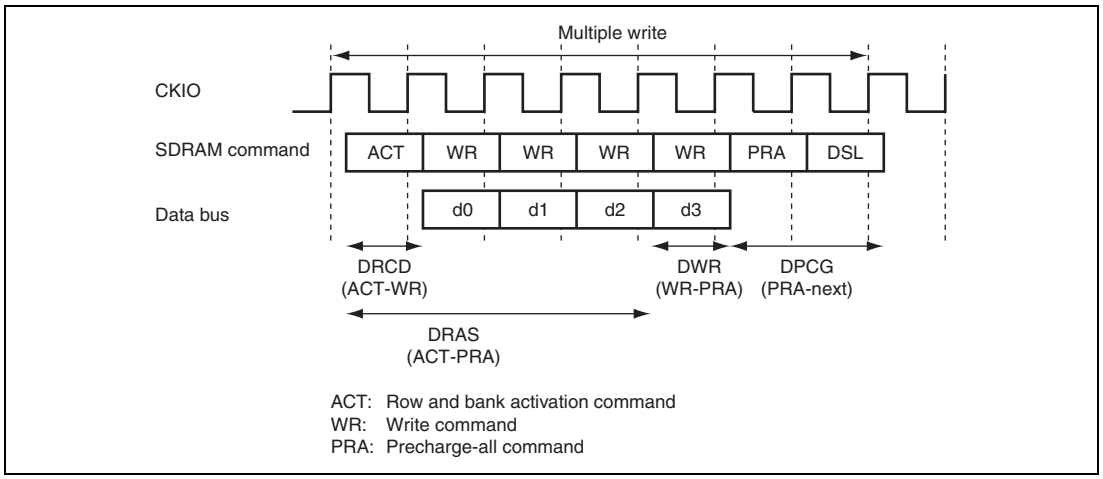


Figure 9.33 Multiple Write Timing Example 1

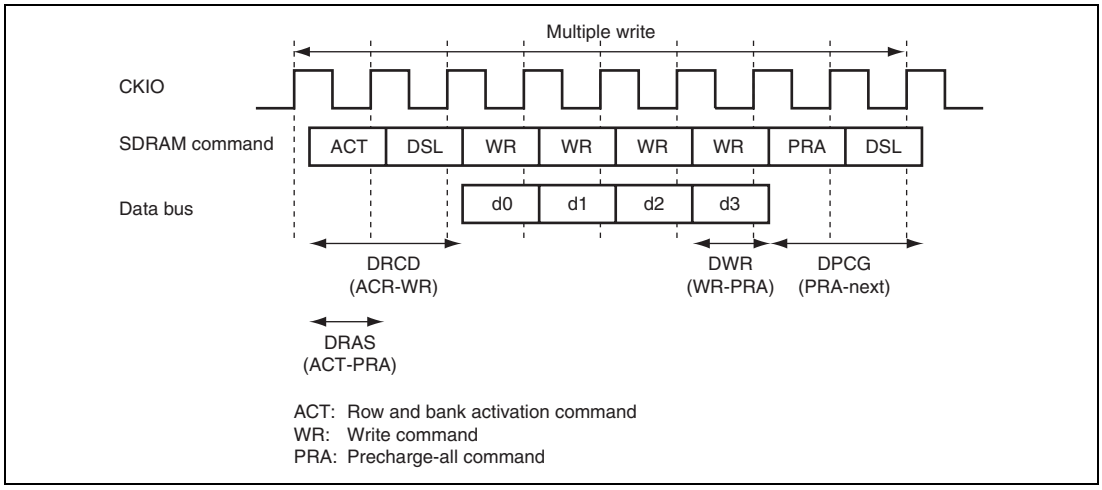


Figure 9.34 Multiple Write Timing Example 2

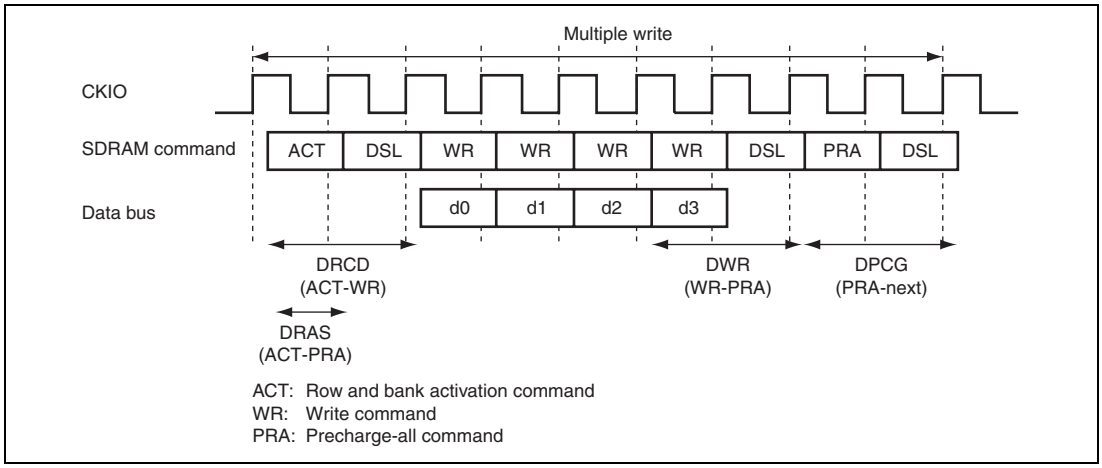


Figure 9.35 Multiple Write Timing Example 3

- Single Read Timing Setting Examples

  Figures 9.36 to 9.38 show the correspondence between the timing of single read operations and the set values of the SDRAMm timing register (SDmTR). Table 9.14 shows the SDRAMm timing register (SDmTR) set values for each figure.

Table 9.14 SDITR Set Value Correspondence Table (Single Read Timing)

Figure	DRAS	DRCD	DPCG	DCL
Figure 9.36	010	00	001	010
Figure 9.37	000	01	001	010
Figure 9.38	000	01	001	011

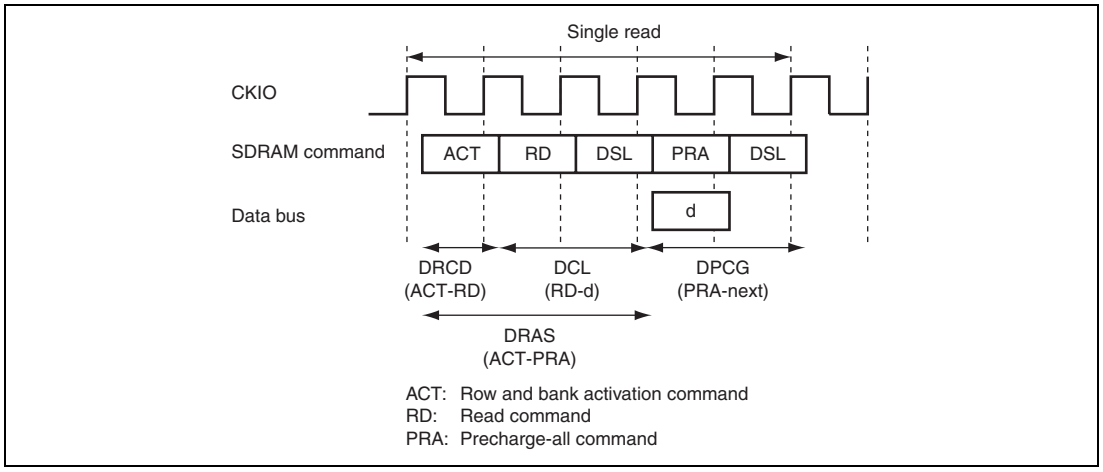
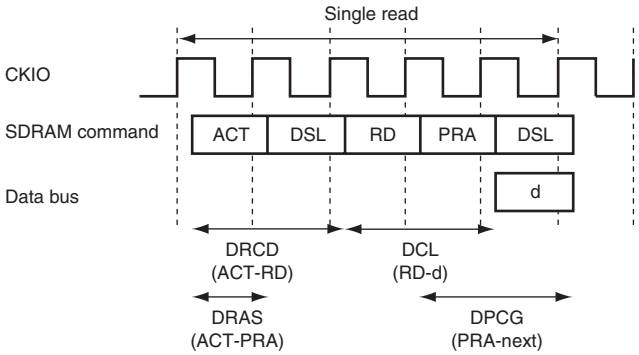


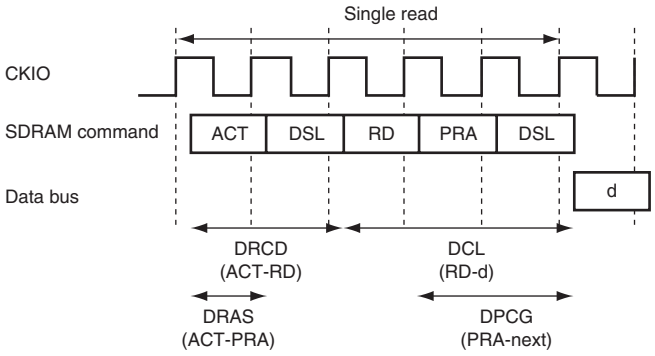
Figure 9.36 Single Read Timing Example 1



ACT: Row and bank activation command
RD: Read command
DSL: Deselect command
PRA: Precharge-all command

Note: If the interval set in DRAS ends before RD, PRA is issued in the table size after RD.

Figure 9.37 Single Read Timing Example 2



ACT: Row and bank activation command
RD: Read command
PRA: Precharge-all command

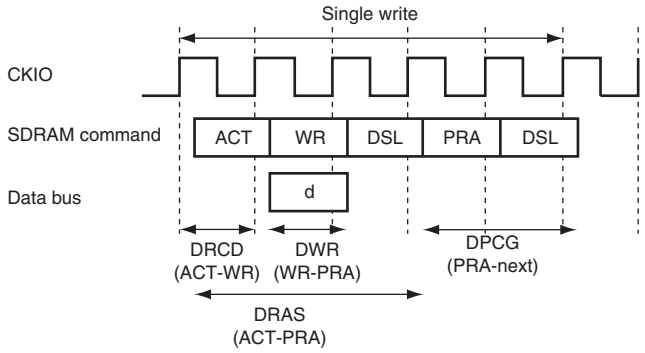
Figure 9.38 Single Read Timing Example 3

- Single Write Timing Setting Examples

  Figures 9.39 to 9.41 show the correspondence between the timing of single write operations and the set values of the SDRAMm timing register (SDmTR). Table 9.15 shows the SDRAMm timing register (SDmTR) set values for each figure.

Table 9.15 SDITR Set Value Correspondence Table (Single Write Timing)

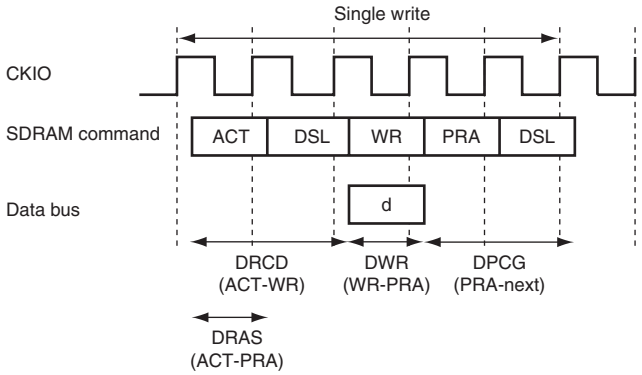
Figure	DRAS	DRCD	DPCG	DWR
Figure 9.39	010	00	001	0
Figure 9.40	000	01	001	0
Figure 9.41	000	01	001	1



ACT: Row and bank activation command
WR: Write command
DSL: Deselect command
PRA: Precharge-all command

Note: If the interval set in DRAS is longer than the period from when the WR command is issued until the DWR interval elapses, the DRAS setting is used.

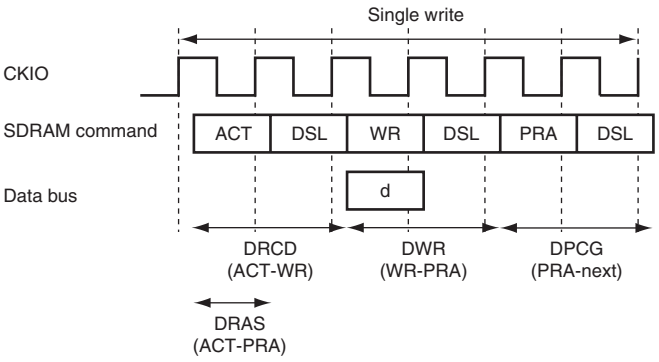
Figure 9.39 Single Write Timing Example 1



ACT: Row and bank activation command
WR: Write command
DSL: Deselect command
PRA: Precharge-all command

Note: If the interval set in DRAS is longer than the period from when the WR command is issued until the DRAS interval elapses, the DWR setting is used.

Figure 9.40 Single Write Timing Example 2



ACT: Row and bank activation command
WR: Write command
DSL: Deselect command
PRA: Precharge-all command

Figure 9.41 Single Write Timing Example 3

(13) External Address/SDRAM Address Signal Multiplex

(a) Address Multiplex

Either of addresses used for accessing external device or SDRAM is output through external address pins. The SDRAM address is shifted internally by changing the settings of DDBW and DSZ in SDmADR and BSIZE in SDCmCNT. The bank address is output on A16 and A15 and the address is output on A14 to A2.

Table 9.16 External Address/SDRAM Address Pins

Pin Name	Function	Pin Name	Function
A27	External address	A13 (/MA11)	External address/SDRAM address
A26	External address	A12 (/MA10)	External address/SDRAM address
A25	External address	A11 (/MA9)	External address/SDRAM address
A24	External address	A10 (/MA8)	External address/SDRAM address
A23	External address	A9 (/MA7)	External address/SDRAM address
A22	External address	A8 (/MA6)	External address/SDRAM address
A21	External address	A7 (/MA5)	External address/SDRAM address
A20	External address	A6 (/MA4)	External address/SDRAM address
A19	External address	A5 (/MA3)	External address/SDRAM address
A18	External address	A4 (/MA2)	External address/SDRAM address
A17	External address	A3 (/MA1)	External address/SDRAM address
A16 (/BA1)	External address/SDRAM bank address	A2 (/MA0)	External address/SDRAM address
A15 (/BA0)	External address/SDRAM bank address	A1	External address
A14 (/MA12)	External address/SDRAM address	A0	External address

(b) Address Register Setting Value and Supported SDRAM Configuration

Tables 9.17 to 9.19 are the SDRAM configurations that to support for 8-, 16-, or 32-bit bus width. These tables are featured to ease the understanding of the relationships between the SDRAM to support and address multiplex.

Addresses addr27 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM. The table below shows how the settings of DSZ and DDBW determine which signals are output on the SDRAM-access pins.

Table 9.17 Case for 8-Bit External Data Bus Width (BSIZE*[1] = (1, 0))

SDRAM Type	64 Mbits (×8)		128 Mbits (×8)		256 Mbits (×8)		512 Mbits (×8)	
Number	1		1		1		1	
DSZ*[2]	001 (8 Mbytes)		010 (16 Mbytes)		011 (32 Mbytes)		100 (64 Mbytes)	
DDBW*[3]	00 (8 bits)		00 (8 bits)		00 (8 bits)		00 (8 bits)	
This LSI address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A16 (/BA1)*[4]	addr22*[5]	addr22*[5]	addr23*[5]	addr23*[5]	addr24*[5]	addr24*[5]	addr25*[5]	addr25*[5]
A15 (/BA0)*[4]	addr21*[5]	addr21*[5]	addr22*[5]	addr22*[5]	addr23*[5]	addr23*[5]	addr24*[5]	addr24*[5]
A14 (/MA12)*[4]	L	L	L	L	addr22*[5]	L	addr23*[5]	L
A13 (/MA11)*[4]	addr20*[5]	L	addr21*[5]	L	addr21*[5]	L	addr22*[5]	addr10*[5]
A12 (/MA10)*[4]	addr19*[5]	*[6]	addr20*[5]	*[6]	addr20*[5]	*[6]	addr21*[5]	*[6]
A11 (/MA9)*[4]	addr18*[5]	L	addr19*[5]	addr9*[5]	addr19*[5]	addr9*[5]	addr20*[5]	addr9*[5]
A10 (/MA8)*[4]	addr17*[5]	addr8*[5]	addr18*[5]	addr8*[5]	addr18*[5]	addr8*[5]	addr19*[5]	addr8*[5]
A9 (/MA7)*[4]	addr16*[5]	addr7*[5]	addr17*[5]	addr7*[5]	addr17*[5]	addr7*[5]	addr18*[5]	addr7*[5]
A8 (/MA6)*[4]	addr15*[5]	addr6*[5]	addr16*[5]	addr6*[5]	add16*[5]	addr6*[5]	addr17*[5]	addr6*[5]
A7 (/MA5)*[4]	addr14*[5]	addr5*[5]	addr15*[5]	addr5*[5]	addr15*[5]	addr5*[5]	addr16*[5]	addr5*[5]
A6 (/MA4)*[4]	addr13*[5]	addr4*[5]	addr14*[5]	addr4*[5]	addr14*[5]	addr4*[5]	addr15*[5]	addr4*[5]
A5 (/MA3)*[4]	addr12*[5]	addr3*[5]	addr13*[5]	addr3*[5]	addr13*[5]	addr3*[5]	addr14*[5]	addr3*[5]
A4 (/MA2)*[4]	addr11*[5]	addr2*[5]	addr12*[5]	addr2*[5]	addr12*[5]	addr2*[5]	addr13*[5]	addr2*[5]
A3 (/MA1)*[4]	addr10*[5]	addr1*[5]	addr11*[5]	addr1*[5]	addr11*[5]	addr1*[5]	addr12*[5]	addr1*[5]
A2 (/MA0)*[4]	addr9*[5]	addr0*[5]	addr10*[5]	addr0*[5]	addr10*[5]	addr0*[5]	addr11*[5]	addr0*[5]

Notes:
1. The legend BSIZE represents the BSIZE bit in the SDRAMCm control register.
2. The legend DSZ represents the DSZ bit in the SDRAMm address register.
3. The legend DDBW represents the DDBW bit in the SDRAMm address register.
4. The legends BA1, BA0, and MA12 to MA0 represent the SDRAM bank address and SDRAM address respectively.
5. Addresses addr25 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM.
6. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 9.18 Case for 16-Bit External Data Bus Width (BSIZE*1 = (0, 0)) (1)

SDRAM Type Number	64 Mbits (×16) 1		64 Mbits (×8) 2		128 Mbits (×16) 1		128 Mbits (×8) 2	
DSZ*2	001 (8 Mbytes)		010 (16 Mbytes)		010 (16 Mbytes)		011 (32 Mbytes)	
DDBW*3	01 (16 bits)		00 (8 bits)		01 (16 bits)		00 (8 bits)	
This LSI address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A16 (/BA1)*4	addr22*5	addr22*5	addr23*5	addr23*5	addr23*5	addr23*5	addr24*5	addr24*5
A15 (/BA0)*4	addr21	addr21*5	addr22*5	addr22*5	addr22*5	addr22*5	addr23*5	addr23*5
A14 (/MA12)*4	L	L	L	L	L	L	L	L
A13 (/MA11)*4	addr20*5	L	addr21*5	L	addr21*5	L	addr22*5	L
A12 (/MA10)*4	addr19*5	*6	addr20*5	*6	addr20*5	*6	addr21*5	*6
A11 (/MA9)*4	addr18*5	L	addr19*5	L	addr19*5	L	addr20*5	addr10*5
A10 (/MA8)*4	addr17*5	L	addr18*5	addr9*5	addr18*5	addr9*5	addr19*5	addr9*5
A9 (/MA7)*4	addr16*5	addr8*5	addr17*5	addr8*5	addr17*5	addr8*5	addr18*5	addr8*5
A8 (/MA6)*4	addr15*5	addr7*5	addr16*5	addr7*5	addr16*5	addr7*5	addr17*5	addr7*5
A7 (/MA5)*4	addr14*5	addr6*5	addr15*5	addr6*5	addr15*5	addr6*5	addr16*5	addr6*5
A6 (/MA4)*4	addr13*5	addr5*5	addr14*5	addr5*5	addr14*5	addr5*5	addr15*5	addr5*5
A5 (/MA3)*4	addr12*5	addr4*5	addr13*5	addr4*5	addr13*5	addr4*5	addr14*5	addr4*5
A4 (/MA2)*4	addr11*5	addr3*5	addr12*5	addr3*5	addr12*5	addr3*5	addr13*5	addr3*5
A3 (/MA1)*4	addr10*5	addr2*5	addr11*5	addr2*5	addr11*5	addr2*5	addr12*5	addr2*5
A2 (/MA0)*4	addr9*5	addr1*5	addr10*5	addr1*5	addr10*5	addr1*5	addr11*5	addr1*5

Notes: 1. The legend BSIZE represents the BSIZE bit in the SDRAMCm control register.
2. The legend DSZ represents the DSZ bit in the SDRAMm address register.
3. The legend DDBW represents the DDBW bit in the SDRAMm address register.
4. The legends BA1, BA0, and MA12 to MA0 represent the SDRAM bank address and SDRAM address respectively.
5. Addresses addr24 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM.
6. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 9.18 Case for 16-Bit External Data Bus Width (BSIZE*[1] = (0, 0)) (2)

SDRAM Type	256 Mbits (×16)		256 Mbits (×8)		512 Mbits (×16)		512 Mbits (×8)	
Number	1		2		1		2	
DSZ*[2]	011 (32 Mbytes)		100 (64 Mbytes)		100 (64 Mbytes)		101 (128 Mbytes)	
DDBW*[3]	01 (16 bits)		00 (8 bits)		01 (16 bits)		00 (8 bits)	
This LSI address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A16 (/BA1)*[4]	addr24*[5]	addr24*[5]	addr25*[5]	addr25*[5]	addr25*[5]	addr25*[5]	addr26*[5]	addr26*[5]
A15 (/BA0)*[4]	addr23*[5]	addr23*[5]	addr24*[5]	addr24*[5]	addr24*[5]	addr24*[5]	addr25*[5]	addr25*[5]
A14 (/MA12)*[4]	addr22*[5]	L	addr23*[5]	L	addr23*[5]	L	addr24*[5]	L
A13 (/MA11)*[4]	addr21*[5]	L	addr22*[5]	L	addr22*[5]	L	addr23*[5]	addr11*[5]
A12 (/MA10)*[4]	addr20*[5]	*[6]	addr21*[5]	*[6]	addr21*[5]	*[6]	addr22*[5]	*[6]
A11 (/MA9)*[4]	addr19*[5]	L	addr20*[5]	addr10*[5]	addr20*[5]	addr10*[5]	addr21*[5]	addr10*[5]
A10 (/MA8)*[4]	addr18*[5]	addr9*[5]	addr19*[5]	addr9*[5]	addr19*[5]	addr9*[5]	addr20*[5]	addr9*[5]
A9 (/MA7)*[4]	addr17*[5]	addr8*[5]	addr18*[5]	addr8*[5]	addr18*[5]	addr8*[5]	addr19*[5]	addr8*[5]
A8 (/MA6)*[4]	addr16*[5]	addr7*[5]	addr17*[5]	addr7*[5]	addr17*[5]	addr7*[5]	addr18*[5]	addr7*[5]
A7 (/MA5)*[4]	addr15*[5]	addr6*[5]	addr16*[5]	addr6*[5]	addr16*[5]	addr6*[5]	addr17*[5]	addr6*[5]
A6 (/MA4)*[4]	addr14*[5]	addr5*[5]	addr15*[5]	addr5*[5]	addr15*[5]	addr5*[5]	addr16*[5]	addr5*[5]
A5 (/MA3)*[4]	addr13*[5]	addr4*[5]	addr14*[5]	addr4*[5]	addr14*[5]	addr4*[5]	addr15*[5]	addr4*[5]
A4 (/MA2)*[4]	addr12*[5]	addr3*[5]	addr13*[5]	addr3*[5]	addr13*[5]	addr3*[5]	addr14*[5]	addr3*[5]
A3 (/MA1)*[4]	addr11*[5]	addr2*[5]	addr12*[5]	addr2*[5]	addr12*[5]	addr2*[5]	addr13*[5]	addr2*[5]
A2 (/MA0)*[4]	addr10*[5]	addr1*[5]	addr11*[5]	addr1*[5]	addr11*[5]	addr1*[5]	addr12*[5]	addr1*[5]

Notes: 1. The legend BSIZE represents the BSIZE bit in the SDRAMCm control register.
2. The legend DSZ represents the DSZ bit in the SDRAMm address register.
3. The legend DDBW represents the DDBW bit in the SDRAMm address register.
4. The legends BA1, BA0, and MA12 to MA0 represent the SDRAM bank address and SDRAM address respectively.
5. Addresses addr26 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM.
6. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 9.19 Case for 32-Bit External Data Bus Width (BSIZE*1 = (0, 1)) (1)

SDRAM Type Number	64 Mbits (×32) 1		64 Mbits (×16) 2		128 Mbits (×32) 1		64 Mbits (×8) 4	
DSZ*2	001 (8 Mbytes)		010 (16 Mbytes)		010 (16 Mbytes)		011 (32 Mbytes)	
DDBW*3	10 (32 bits)		01 (16 bits)		10 (32 bits)		00 (8 bits)	
This LSI address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A16 (/BA1)*4	addr22*5	addr22*5	addr23*5	addr23*5	addr23*5	addr23*5	addr24*5	addr24*5
A15 (/BA0)*4	addr21*5	addr21*5	addr22*5	addr22*5	addr22*5	addr22*5	addr23*5	addr23*5
A14 (/MA12)*4	L	L	L	L	L	L	L	L
A13 (/MA11)*4	L	L	addr21*5	L	addr21*5	L	addr22*5	L
A12 (/MA10)*4	addr20*5	*6	addr20*5	*6	addr20*5	*6	addr21*5	*6
A11 (/MA9)*4	addr19*5	L	addr19*5	L	addr19*5	L	addr20*5	L
A10 (/MA8)*4	addr18*5	L	addr18*5	L	addr18*5	L	addr19*5	addr10*5
A9 (/MA7)*4	addr17*5	addr9*5	addr17*5	addr9*5	addr17*5	addr9*5	addr18*5	addr9*5
A8 (/MA6)*4	addr16*5	addr8*5	addr16*5	addr8*5	addr16*5	addr8*5	addr17*5	addr8*5
A7 (/MA5)*4	addr15*5	addr7*5	addr15*5	addr7*5	addr15*5	addr7*5	addr16*5	addr7*5
A6 (/MA4)*4	addr14*5	addr6*5	addr14*5	addr6*5	addr14*5	addr6*5	addr15*5	addr6*5
A5 (/MA3)*4	addr13*5	addr5*5	addr13*5	addr5*5	addr13*5	addr5*5	addr14*5	addr5*5
A4 (/MA2)*4	addr12*5	addr4*5	addr12*5	addr4*5	addr12*5	addr4*5	addr13*5	addr4*5
A3 (/MA1)*4	addr11*5	addr3*5	addr11*5	addr3*5	addr11*5	addr3*5	addr12*5	addr3*5
A2 (/MA0)*4	addr10*5	addr2*5	addr10*5	addr2*5	addr10*5	addr2*5	addr11*5	addr2*5

Notes: 1. The legend BSIZE represents the BSIZE bit in the SDRAMCm control register.
2. The legend DSZ represents the DSZ bit in the SDRAMm address register.
3. The legend DDBW represents the DDBW bit in the SDRAMm address register.
4. The legends BA1, BA0, and MA12 to MA0 represent the SDRAM bank address and SDRAM address respectively.
5. Addresses addr24 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM.
6. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 9.19 Case for 32-Bit External Data Bus Width (BSIZE*1 = (0, 1)) (2)

SDRAM Type	128 Mbits (×16)		256 Mbits (×32)		128 Mbits (×8)		256 Mbits (×16)	
Number	2		1		4		2	
DSZ*2	011 (32 Mbytes)		011 (32 Mbytes)		100 (64 Mbytes)		100 (64 Mbytes)	
DDBW*3	01 (16 bits)		10 (32 bits)		00 (8 bits)		01 (16 bits)	
This LSI address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A16 (/BA1)*4	addr24*5	addr24*5	addr24*5	addr24*5	addr25*5	addr25*5	addr25*5	addr25*5
A15 (/BA0)*4	addr23*5	addr23*5	addr23*5	addr23*5	addr24*5	addr24*5	addr24*5	addr24*5
A14 (/MA12)*4	L	L	L	L	L	L	addr23*5	L
A13 (/MA11)*4	addr22*5	L	addr22*5	L	addr23*5	L	addr22*5	L
A12 (/MA10)*4	addr21*5	*6	addr21*5	*6	addr22*5	*6	addr21*5	*6
A11 (/MA9)*4	addr20*5	L	addr20*5	L	addr21*5	addr11*5	addr20*5	L
A10 (/MA8)*4	addr19*5	addr10*5	addr19*5	addr10*5	addr20*5	addr10*5	addr19*5	addr10*5
A9 (/MA7)*4	addr18*5	addr9*5	addr18*5	addr9*5	addr19*5	addr9*5	addr18*5	addr9*5
A8 (/MA6)*4	addr17*5	addr8*5	addr17*5	addr8*5	addr18*5	addr8*5	addr17*5	addr8*5
A7 (/MA5)*4	addr16*5	addr7*5	addr16*5	addr7*5	addr17*5	addr7*5	addr16*5	addr7*5
A6 (/MA4)*4	addr15*5	addr6*5	addr15*5	addr6*5	addr16*5	addr6*5	addr15*5	addr6*5
A5 (/MA3)*4	addr14*5	addr5*5	addr14*5	addr5*5	addr15*5	addr5*5	addr14*5	addr5*5
A4 (/MA2)*4	addr13*5	addr4*5	addr13*5	addr4*5	addr14*5	addr4*5	addr13*5	addr4*5
A3 (/MA1)*4	addr12*5	addr3*5	addr12*5	addr3*5	addr13*5	addr3*5	addr12*5	addr3*5
A2 (/MA0)*4	addr11*5	addr2*5	addr11*5	addr2*5	addr12*5	addr2*5	addr11*5	addr2*5

Notes: 1. The legend BSIZE represents the BSIZE bit in the SDRAMCm control register.
2. The legend DSZ represents the DSZ bit in the SDRAMm address register.
3. The legend DDBW represents the DDBW bit in the SDRAMm address register.
4. The legends BA1, BA0, and MA12 to MA0 represent the SDRAM bank address and SDRAM address respectively.
5. Addresses addr25 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM.
6. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

Table 9.19 Case for 32-Bit External Data Bus Width (BSIZE*1 = (0, 1)) (3)

SDRAM Type	512 Mbits (×32)		256 Mbits (×8)		512 Mbits (×16)		512 Mbits (×8)	
Number	1		4		2		4	
DSZ*2	100 (64 Mbytes)		101 (128 Mbytes)		101 (128 Mbytes)		110 (256 Mbytes)	
DDBW*3	10 (32 bits)		00 (8 bits)		01 (16 bits)		00 (8 bits)	
This LSI address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address	Row Address	Column Address
A16 (/BA1)*4	addr25*5	addr25*5	addr26*5	addr26*5	addr26*5	addr26*5	addr27*5	addr27*5
A15 (/BA0)*4	addr24*5	addr24*5	addr25*5	addr25*5	addr25*5	addr25*5	addr26*5	addr26*5
A14 (/MA12)*4	addr23*5	L	addr24*5	L	addr24*5	L	addr25*5	L
A13 (/MA11)*4	addr22*5	L	addr23*5	L	addr23*5	L	addr24*5	addr12*5
A12 (/MA10)*4	addr21*5	*6	addr22*5	*6	addr22*5	*6	addr23*5	*6
A11 (/MA9)*4	addr20*5	L	addr21*5	addr11*5	addr21*5	addr11*5	addr22*5	addr11*5
A10 (/MA8)*4	addr19*5	addr10*5	addr20*5	addr10*5	addr20*5	addr10*5	addr21*5	addr10*5
A9 (/MA7)*4	addr18*5	addr9*5	addr19*5	addr9*5	addr19*5	addr9*5	addr20*5	addr9*5
A8 (/MA6)*4	addr17*5	addr8*5	addr18*5	addr8*5	addr18*5	addr8*5	addr19*5	addr8*5
A7 (/MA5)*4	addr16*5	addr7*5	addr17*5	addr7*5	addr17*5	addr7*5	addr18*5	addr7*5
A6 (/MA4)*4	addr15*5	addr6*5	addr16*5	addr6*5	addr16*5	addr6*5	addr17*5	addr6*5
A5 (/MA3)*4	addr14*5	addr5*5	addr15*5	addr5*5	addr15*5	addr5*5	addr16*5	addr5*5
A4 (/MA2)*4	addr13*5	addr4*5	addr14*5	addr4*5	addr14*5	addr4*5	addr15*5	addr4*5
A3 (/MA1)*4	addr12*5	addr3*5	addr13*5	addr3*5	addr13*5	addr3*5	addr14*5	addr3*5
A2 (/MA0)*4	addr11*5	addr2*5	addr12*5	addr2*5	addr12*5	addr2*5	addr13*5	addr2*5

Notes: 1. The legend BSIZE represents the BSIZE bit in the SDRAMCm control register.
2. The legend DSZ represents the DSZ bit in the SDRAMm address register.
3. The legend DDBW represents the DDBW bit in the SDRAMm address register.
4. The legends BA1, BA0, and MA12 to MA0 represent the SDRAM bank address and SDRAM address respectively.
5. Addresses addr27 to addr0 are the logical addresses used by the CPU and DMAC in access to the SDRAM.
6. When the RD, WR or PRA command is issued, this carries the pre-charge option signal.

(c) Example of SDRAM Connection

Figures 9.42 and 9.43 show examples of the connection of SDRAM with this LSI.

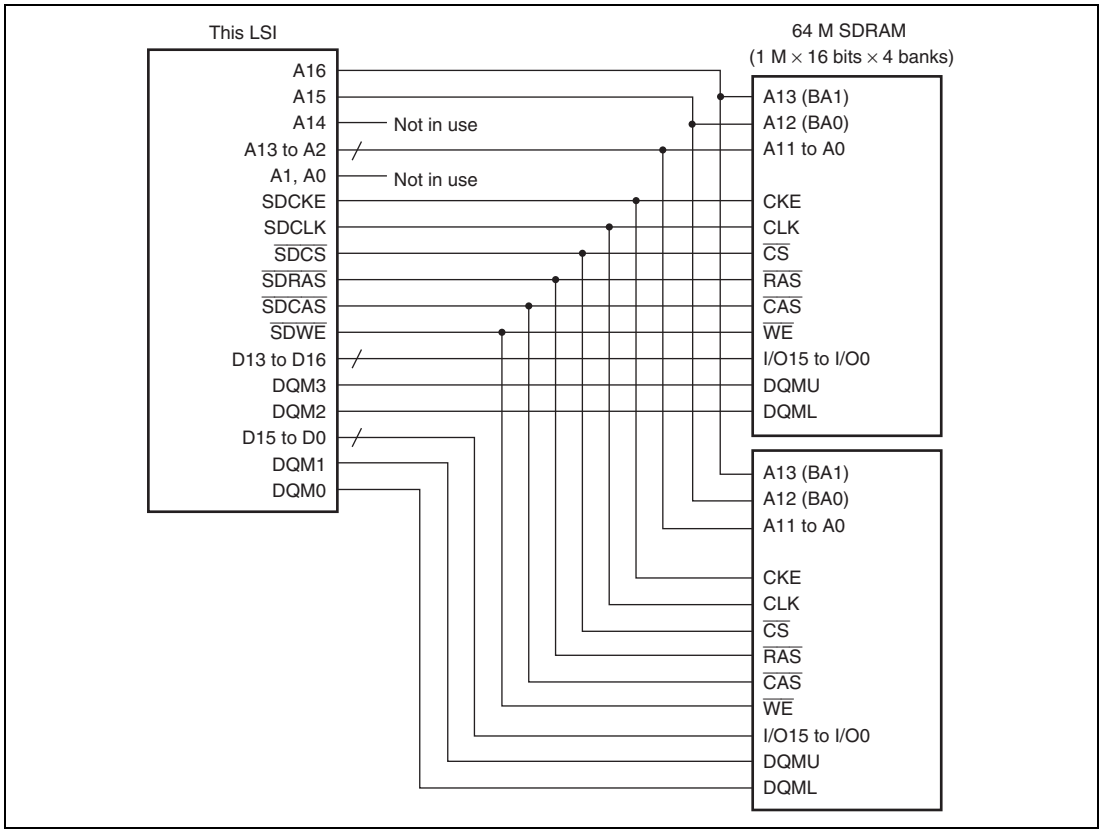


Figure 9.42 Example of Connecting a 32-Bit Data-Width SDRAM

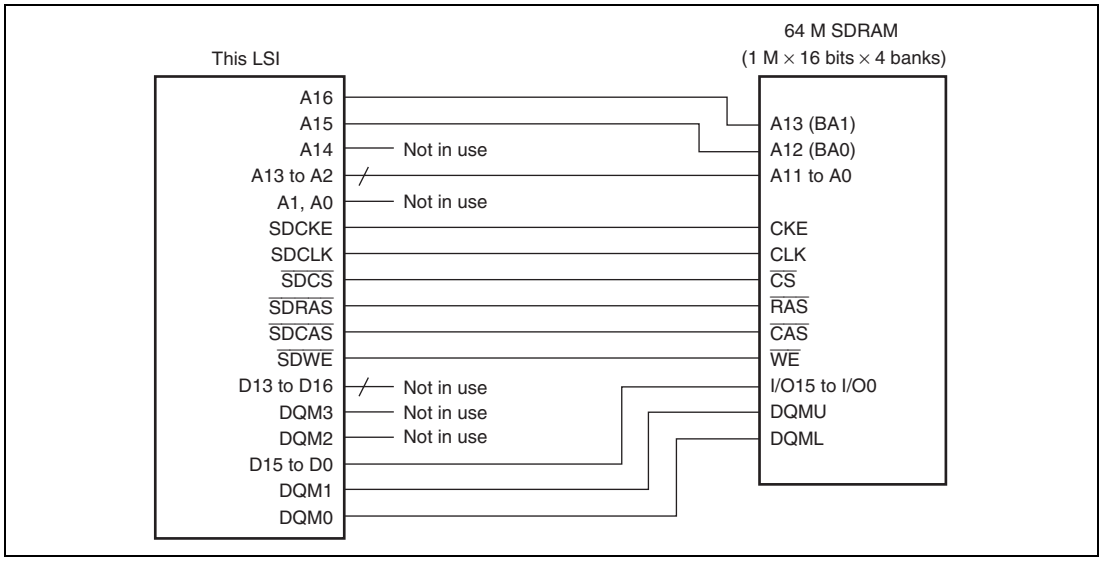


Figure 9.43 Example of Connecting a 16-Bit Data-Width SDRAM

9.6 Usage Note

9.6.1 Note on Power-on Reset Exception Handling and Deep Standby Mode Cancellation

When writing to the external address space or making SDRAM settings in power-on reset exception handling or cancellation of deep standby mode, be sure to set bits ACOSW[3:0] in ACSWR to B'0011 beforehand.

9.6.2 Write Buffer

In write access to normal or SDRAM space, the write data are stored once in the internal write buffer of the BSC, and only after that does actual writing to the device (external device) connected in the normal or SDRAM space proceed. Since writing from the write buffer to the external device is performed automatically, no processing by software is necessary.

However, care must be taken on the following point. Write access from the CPU or DMAC appears complete at the point where the data are stored in the above write buffer. That is, at the point where the write access from the CPU or DMA controller has been completed, writing to the external device might not have been completed. To confirm the completion of writing to the external device, dummy read the normal or SDRAM space. Completion of the dummy-read operation guarantees the completion of writing to the external device in response to previous write access. The target address for the dummy read operation does not have to be in the same device as the target for write access. Furthermore, it does not have to be in the same space.

9.6.3 Note on Transition to Software Standby Mode or Deep Standby Mode

When a transition to software standby mode or deep standby mode is made after write access to the normal or SDRAM space, there is a possibility that data remains in the internal write buffer of the BSC. To confirm that no data remain in the write buffer, execute a dummy read of the external device in the same way as described above.

Section 10 Bus Monitor

The bus monitor is a module that monitors bus errors on each bus. When an illegal address access or a bus timeout is detected, a bus error interrupt is generated and an access canceling signal is output for the bus timeout. (The bus timeout function is used for debugging.)

Figure 10.1 shows a block diagram of the bus monitor.

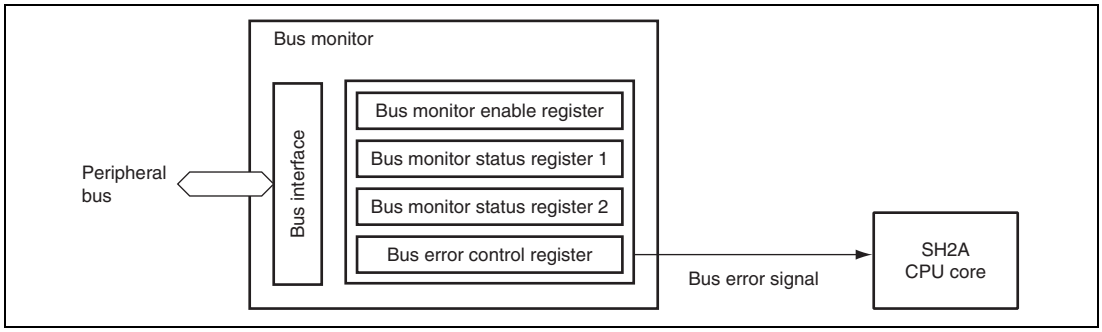


Figure 10.1 Block Diagram of Bus Monitor

10.1 Register Descriptions

The bus monitor has the following registers.

All registers are initialized by a power-on reset or in deep standby mode.

Table 10.1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bus monitor enable register	SYCBEEN	R/W	H'00	H'FF400000	8, 16, 32
Bus monitor status register 1	SYCBESTS1	R/W	H'00	H'FF400004	8, 16, 32
Bus monitor status register 2	SYCBESTS2	R/W	H'00	H'FF400008	8, 16, 32
Bus error control register	SYCBESW	R/W	H'00	H'FF40000C	8, 16, 32

10.1.1 Bus Monitor Enable Register (SYCBEEN)

SYCBEEN clears the bus monitor status register and controls the detection function.

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

Bit	Bit Name	Initial Value	R/W	Description
31	STSCLR	0	R/W	Status Clear Writing 1 to this bit clears the bus monitor status register. Writing 0 or reading data has no effect. 0: Invalid 1: Bus monitor status register cleared
30 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18	TOEN	0	R/W	Timeout Detection Enable This bit enables or disables the function that detects a bus timeout on each bus. 0: Bus timeout detection function disabled 1: Bus timeout detection function enabled
17	IGAEN	0	R/W	Illegal Address Access Detection Enable This bit enables or disables the function that detects an illegal address access on each bus. 0: Illegal address access detection function disabled 1: Illegal address access detection function enabled
16 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: When a bus access is performed with the detection function disabled (TOEN = 0), the bus may freeze.

10.1.2 Bus Monitor Status Register 1 (SYCBESTS1)

SYCBESTS1 indicates the status of slave bus (peripheral bus (1)) regarding whether a timeout occurred, whether an illegal address access was made, or which bus master accessed the slave bus. Table 10.2 shows the correspondence between the bus space and the slave bus.

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
	—	PTO	PER	—	—	—	PMST[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

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	PTO	0	R	Timeout This bit indicates that a timeout occurred on peripheral bus (1) when the first bus error occurred. 0: Timeout not generated 1: Timeout generated
13	PER	0	R	Illegal Address Access This bit indicates that an illegal address access was made on peripheral bus (1) when the first bus error occurred. 0: Illegal address access not made 1: Illegal address access made
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9, 8	PMST[1:0]	00	R	Bus Master These bits indicate the bus master that accessed peripheral bus (1) when the first bus error occurred. 00: CPU 01: DMAC (destination side) 10: Setting prohibited 11: DMAC (source side)
7 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 10.2 Bus Space and Slave Bus

Address	Bus Space	Slave Bus
H'0000 0000 to H'4FFF FFFF	External bus space	External bus
H'5000 0000 to H'E800 FFFF	Reserved	(Others*[1])
H'E801 0000 to H'EFFF FFFF	Reserved	(Others*[1])
H'F000 0000 to H'F1FF FFFF	Address array space in cache	—*[2]
H'F200 0000 to H'F5FF FFFF	Reserved	—*[2]
H'F600 0000 to H'FF3F FFFF	Reserved	(Others*[1])
H'FF40 0000 to H'FF5F FFFF	On-chip peripheral module (1)	Peripheral bus (1)
H'FF60 0000 to H'FFF7 FFFF	Reserved	(Others*[1])
H'FFF8 0000 to H'FFF8 7FFF	On-chip RAM	—*[2]
H'FFF8 8000 to H'FFFB FFFF	Reserved	—*[2]
H'FFFC 0000 to H'FFFF FFFF	On-chip peripheral module (2)	Peripheral bus (2)

Notes: 1. This means bus spaces in the slave bus space other than those for the external bus and peripheral buses (1) and (2).
2. An illegal address access error does not occur.

10.1.3 Bus Monitor Status Register 2 (SYCBESTS2)

SYCBESTS2 indicates the status of slave buses (external bus/peripheral bus (2)/others) regarding whether a timeout occurred, whether an illegal address access was made, or which bus master accessed the slave bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ETO	EER	—	—	—	EMST[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

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OER	—	—	—	OMST[1:0]		—	—	SHER	—	—	—	SHMST[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

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30	ETO	0	R	Timeout This bit indicates that a timeout occurred on the external bus when the first bus error occurred. 0: Timeout not generated 1: Timeout generated
29	EER	0	R	Illegal Address Access This bit indicates that an illegal address access was made on the external bus when the first bus error occurred. 0: Illegal address access not made 1: Illegal address access made
28 to 26	—	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
25, 24	EMST[1:0]	00	R	Bus Master These bits indicate the bus master that accessed the external bus when the first bus error occurred. 00: CPU 01: DMAC (destination side) 10: Setting prohibited 11: DMAC (source side)
23 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	OER	0	R	Illegal Address Access These bits indicate the bus master that accessed other buses when the first bus error occurred. 0: Illegal address access not made 1: Illegal address access made
12 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	OMST[1:0]	00	R/W	Bus Master These bits indicate the bus master that accessed other buses when the first bus error occurred. 00: CPU 01: DMAC (destination side) 10: Setting prohibited 11: DMAC (source side)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	SHER	0	R	Illegal Address Access This bit indicates that an illegal address access was made on peripheral bus (2) when the first bus error occurred. 0: Illegal address access not made 1: Illegal address access made

Bit	Bit Name	Initial Value	R/W	Description
4 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	SHMST [1:0]	00	R	Bus Master These bits indicate the bus master that accessed peripheral bus (2) when the first bus error occurred. 00: CPU 01: DMAC (destination side) 10: Setting prohibited 11: DMAC (source side)

10.1.4 Bus Error Control Register (SYCBESW)

SYCBESW controls the notification of various types of bus errors to the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	00CPEN	01CPEN	—	11CPEN	—	—	—	—	—	—	—	—	—	—	—	—
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/W	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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	00CPEN	0	R/W	Bus Error Control (CPU → CPU) This bit controls notification to the CPU when a bus error is caused by the CPU. 0: Not notified 1: Notified
30	01CPEN	0	R/W	Bus Error Control (DMAC Destination Side → CPU) This bit controls notification to the CPU when a bus error is caused by the DMAC destination side. 0: Not notified 1: Notified
29	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
28	11CPEN	0	R/W	Bus Error Control (DMAC Source Side → CPU) This bit controls notification to the CPU when a bus error is caused by the DMAC source side. 0: Not notified 1: Notified
27 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.2 Bus Monitor Function

The bus monitor function detects two types of bus error: illegal address access and bus timeout. Bus error detection is performed in one bus access.

Even when data is transferred in multiple bus accesses such as burst transfer, a bus error can be detected in one bus access.

10.2.1 Operation when a Bus Error is Detected

When a bus error is detected, the status is saved in the bus monitor status register 1 (SYCBESTS1) and bus monitor status register 2 (SYCBESTS2) and the CPU is notified of the bus error is notified to the CPU.

(1) Saving Status in Bus Monitor Status Register or Bus Monitor Status Register 2

When a bus error occurs, the status at the time (what type of error occurred and which bus was being accessed by which bus master) is saved in the bus monitor status register 1 (SYCBESTS1) or bus monitor status register 2 (SYCBESTS2).

Even if another bus error occurs after this, the value in the bus monitor status register (SYCBESTS) or bus monitor status register 2 (SYCBESTS2) is not updated. When multiple bus errors occur at the same time, multiple status bits may be set.

The bus monitor status register 1 (SYCBESTS1) or bus monitor status register 2 (SYCBESTS2) can be cleared by writing 1 to the status clear bit (STSCLR) in the bus monitor enable register (SYCBEEN) from the bus master. After being cleared, the status of a bus error, if generated, is saved in the bus monitor status register 1 (SYCBESTS1) or bus monitor status register 2 (SYCBESTS2) again.

When a clear operation and a bus error happen at the same time, the clear operation has priority and the bus error is ignored.

(2) Error Notification to the CPU

The CPU is notified of a bus error through the OR condition of the timeout bits (PTO/ETO) and illegal address access bits (PER/EER/OER/SHER) in the bus monitor status register 1 (SYCBESTS1) and bus monitor status register 2 (SYCBESTS2). The CPU is notified of a bus error interrupt according to the setting of the bus error control register (SYCBESW).

When the bus monitor status register 1 (SYCBESTS1) and bus monitor status register 2 (SYCBESTS2) are cleared by the CPU, the bus error interrupt signal is also negated.

(3) Termination of Bus Access

When a bus error is detected, the bus access is terminated. For details, see section 10.2.4, Combinations of Masters and Bus Errors.

For the detailed operations when each type of error is detected, see section 10.2.2, Illegal Address Access Detection Function and section 10.2.3, Bus Timeout Detection Function.

10.2.2 Illegal Address Access Detection Function

The illegal address access detection function detects attempted accesses to illegal addresses.

(1) Conditions of Illegal Address Access Error Generation

Illegal address access errors occur when the following illegal addresses are accessed.

- External spaces for which the operation enable bit (EXENB) in the control register of the BSC is not set to "operation enabled"
- Other address areas that are not mapped to any slave bus
- Address areas that are mapped to the slave buses but do not correspond to slave devices

Tables 10.3 and 10.4 show the address areas to which slave devices are not mapped within the spaces for peripheral buses (1) and (2).

Table 10.3 Address Areas without Slave Devices in the Space for Peripheral Bus (1)

FF401000 to FF41FFFF
FF423000 to FF45FFFF
FF464000 to FF5FFFFF

Table 10.4 Address Areas without Slave Devices in the Space for Peripheral Bus (2)

FFFC0000 to FFFDFFFF
FFFE0020 to FFFE03FF
FFFE0420 to FFFE07FF
FFFE0900 to FFFE37FF
FFFE3830 to FFFE387F
FFFE3910 to FFFE3FFF
FFFE4400 to FFFE53FF
FFFE5410 to FFFE57FF
FFFE5840 to FFFE67FF
FFFE6804 to FFFE7FFF
FFFE8100 to FFFE87FF
FFFE8900 to FFFE8FFF
FFFE9100 to FFFE97FF
FFFE9900 to FFFE9FFF
FFFEA100 to FFFEA7FF
FFFEA900 to FFFEAFFF
FFFEB100 to FFFEB7FF
FFFEB900 to FFFECFFF
FFFED010 to FFFED07F
FFFED090 to FFFEDFFF
FFFEE010 to FFFEE07F
FFFEE090 to FFFEE0FF
FFFEE110 to FFFEFFFF
FFFF1408 to FFFF14FF
FFFF1508 to FFFF15FF
FFFF1608 to FFFF16FF
FFFF1720 to FFFF17FF
FFFF1820 to FFFF18FF
FFFF1910 to FFFFFFFF

10.2.3 Bus Timeout Detection Function

The bus timeout detection function detects bus accesses whose cycles are extended to 768 cycles or more.

(1) Conditions of Bus Timeout Error Generation

Bus timeout errors occur in the following cases. This function should be used when debugging software.

- A bus access is not completed on peripheral bus (1)
- The $\overline{\text{WAIT}}$ signal remains asserted during an external bus access

(2) Operation When a Bus Timeout Error is Generated

The operation when a bus timeout error occurs is explained below.

1. The timeout counter starts counting from the next cycle after the start of a bus access.
2. If the bus access is not completed in 768 cycles, a bus timeout occurs and an access canceling signal is asserted for 256 cycles.
   Bus signals such as address, data, BC, read/write, and burst are held.
   The timeout error is recorded in the bus monitor status register 1 (SYCBESTS1) or bus monitor status register 2 (SYCBESTS2).
   A bus error interrupt is generated and sent to the CPU.
3. The bus access is terminated.
4. The CPU processes the bus error.
   Locked buses are all released.

(3) Bus Timeout Operation in Consecutive Accesses

For transfers where multiple bus accesses are made (such as burst transfer), the next bus access might not be terminated when a bus timeout occurs. In this case, a bus timeout may occur continuously.

Even if a bus timeout occurs continuously, the timeout process of terminating a bus access is performed in the same way as the first time. However, the status is saved in the bus monitor status register 1 (SYSCESTS1) or bus monitor status register 2 (SYCBESTS2) only the first time.

10.2.4 Combinations of Masters and Bus Errors

The types of detectable bus error depend on the master and access mode.

(1) CPU Transfer Modes and Types of Bus Error Generated

Table 10.5 shows the types of bus error that may be generated by accesses from the CPU.

Table 10.5 CPU Access Types and Types of Bus Error Generated

Access Type	Normal Access	Burst Access
Illegal address access*1	O*2	O*2*3
Bus timeout*1	O*2	O*2*3

[Legend]

O: A bus error is generated.

—: A bus error is not generated.

Notes: 1. To enable bus error detection, the bus monitor enable register (SYCBEEN) should be set.

2. To notify the CPU of a bus error, the 00CPEN bit in the bus error control register (SYCBESW) should be set to 1.

3. The number of bus errors detected is the same as the number of accesses that resulted in an error.

(2) DMAC Transfer Modes and Operations of Each Bus

Table 10.6 shows the DMAC transfer modes and the types of bus error that may be generated by accesses from the DMAC.

Table 10.6 DMAC Transfer Modes and Types of Bus Error Generated

DMAC Transfer Mode	Cycle Steal	Pipeline
Illegal address access*	O	O
Bus timeout*	O	O

[Legend]
O: A bus error is generated.
—: A bus error is not generated.
Note: * To enable bus error detection, the bus monitor enable register (SYCBEEN) should be set.

10.3 Usage Note

10.3.1 Operation when the CPU is Not Notified of a Bus Error

Table 10.7 describes the operations when bus error notification to the CPU is disabled with the bus error detection enabled (by the setting of the bus monitor enable register (SYCBEEN)).

Table 10.7 Operation When the Master is Not Notified of a Bus Error

Illegal address access	Illegal address access errors equal in number to the predetermined number of transfers are generated and the access is terminated each time.
Bus timeout	Bus timeouts equal in number to the predetermined number of transfers are generated and the access is terminated each time.

Section 11 Direct Memory Access Controller (DMAC)

The DMA controller (hereafter DMAC) is a module that handles high-speed data transfer without CPU intervention in response to requests from software, on-chip peripheral I/O modules, or external pins (external modules). The DMAC itself does not distinguish between requests from on-chip peripheral I/O or external pins (external modules). The DMA supports data transfer between memory units, memory and I/O modules, and I/O modules.

11.1 Features

- Channel number: Up to eight channels (with four channels capable of external requests)
- Transfer requests: Requests from 37 sources including software trigger, on-chip peripheral I/O, and external pins (external modules)
- Maximum transfer bytes: 64 Mbytes
- Address space: 4 Gbytes
- Transfer data sizes:
  - Single data transfer: 8, 16, 32, 64, and 128 bits
  - Single operand transfer: 1, 2, 4, 8, 16, 32, 64, and 128 data
  - Non-stop transfer: Up to the byte count "0"
- Transfer mode:
  - Cycle-stealing transfer (dual-address transfer)
  - Pipelined transfer (dual-address transfer)
- Maximum transfer speed:
  - Cycle-stealing transfer: Minimum of three clock cycles per unit data transfer
  - Pipelined transfer: Minimum of one clock cycle per unit data transfer
- Transfer conditions:
  - Unit operand transfer: a single sequence of single operand data transfer in response to a DMA request
  - Sequential operand transfer: single operand transfers are repeated until the byte count reaches "0"
  - Non-stop transfer: data is continuously transferred until the byte count reaches "0" in response to a single DMA request
- Channel priority:
  Channel 0 > channel 1 > → > channel 6 > channel 7 (this priority order is fixed)

- Interrupt request
  - Two types of interrupt requests (generated when the byte count reaches "0")
    - Interrupt request signals for each channel
    - Interrupt request signal common to all channels
- Reload function (source address, destination address, byte count) settable
- Rotate function settable
- DMAC stop/restart/suspend function settable

Notes: Terminologies in this section are as follows:

1. Single data transfer: Transfer in one read cycle and one write cycle by the DMAC (in the case of dual address transfer)
2. Single operand transfer: Continuous data transfer by the DMAC on one channel (amount of data to be transferred is set in a register)
3. One DMA transfer: Transferring a number of data, from the start address to the end address set in the byte count register
4. Channel number: n = 0 to 7
5. Request source number: k = 1 to 36, m = 0 to 36
6. BIU: Bus Interface Unit (peripheral module). One of the following four kinds according to the source or destination of transfer.
   BIU_E: External space (normal space and SDRAM space)
   BIU_P: Peripheral bus (1) (see figure 1.1), on-chip RAM space
   BIU_SH: Peripheral bus (2) (see figure 1.1)

Figure 11.1 is a block diagram of the DMAC

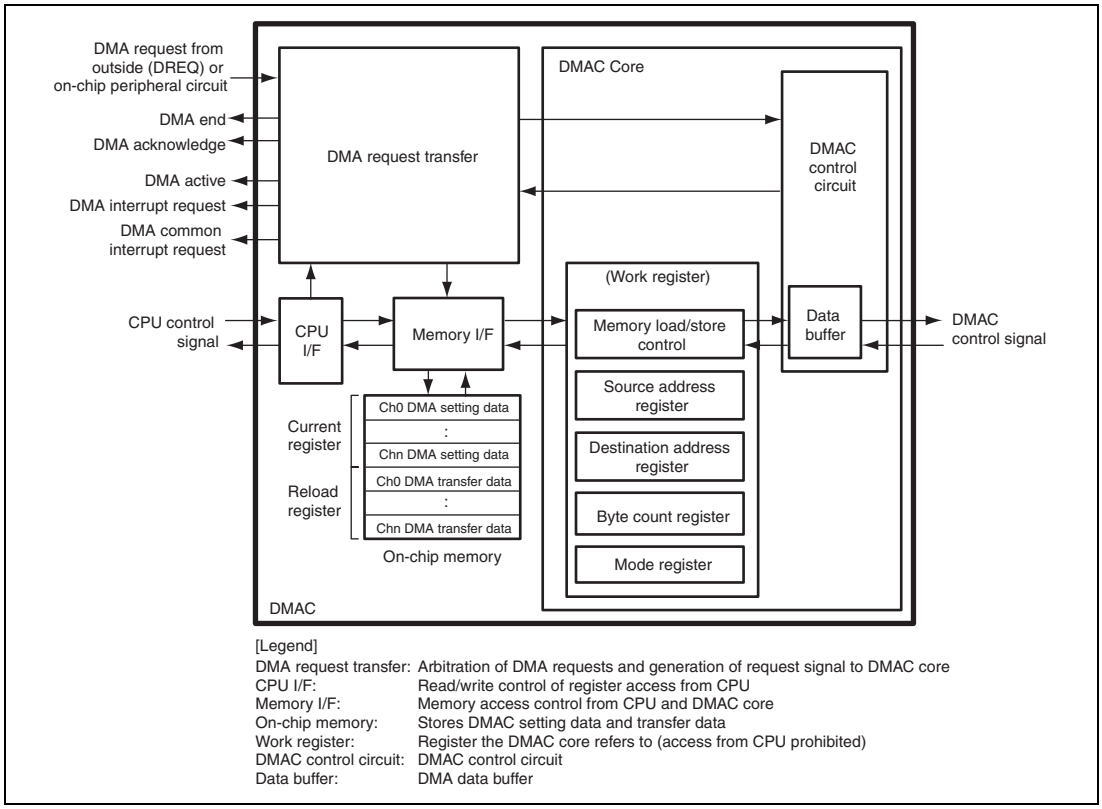


Figure 11.1 DMAC Block Diagram

11.2 Input/Output Pins

Table 11.1 Pin Configuration

Name	I/O	Function
DREQm (m = 0 to 3)	Input	External request for DMA transfer
DACKm (m = 0 to 3)	Output	DMA acknowledgement of external request for DMA transfer (active low)
DACTm (m = 0 to 3)	Output	DMA active in externally requested DMA transfer (active low)
DTENDm (m = 0 to 3)	Output	Completion of externally requested DMA transfer (active low)

11.3 Register Descriptions

The DMAC has the following registers. All registers are initialized by a power-on reset or in deep standby mode.

Table 11.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA current source address register 0	DMCSADR0	R/W	Undefined	H'FF460000	32
	DMA current destination address register 0	DMCDADR0	R/W	Undefined	H'FF460004	32
	DMA current byte count register 0	DMCBCT0	R/W	Undefined	H'FF460008	32
	DMA mode register 0	DMMOD0	R/W	Undefined	H'FF46000C	32
	DMA reload source address register 0	DMRSADR0	R/W	Undefined	H'FF460200	32
	DMA reload destination address register 0	DMRDADR0	R/W	Undefined	H'FF460204	32
	DMA reload byte count register 0	DMRBCT0	R/W	Undefined	H'FF460208	32
	DMA control register A0	DMCNTA0	R/W	H'00000000	H'FF460400	8, 16, 32
	DMA control register B0	DMCNTB0	R/W	H'00000000	H'FF460404	8, 16, 32
1	DMA current source address register 1	DMCSADR1	R/W	Undefined	H'FF460010	32
	DMA current destination address register 1	DMCDADR1	R/W	Undefined	H'FF460014	32
	DMA current byte count register 1	DMCBCT1	R/W	Undefined	H'FF460018	32
	DMA mode register 1	DMMOD1	R/W	Undefined	H'FF46001C	32
	DMA reload source address register 1	DMRSADR1	R/W	Undefined	H'FF460210	32
	DMA reload destination address register 1	DMRDADR1	R/W	Undefined	H'FF460214	32
	DMA reload byte count register 1	DMRBCT1	R/W	Undefined	H'FF460218	32
	DMA control register A1	DMCNTA1	R/W	H'00000000	H'FF460408	8, 16, 32
	DMA control register B1	DMCNTB1	R/W	H'00000000	H'FF46040C	8, 16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
2	DMA current source address register 2	DMCSADR2	R/W	Undefined	H'FF460020	32
	DMA current destination address register 2	DMCDADR2	R/W	Undefined	H'FF460024	32
	DMA current byte count register 2	DMCBCT2	R/W	Undefined	H'FF460028	32
	DMA mode register 2	DMMOD2	R/W	Undefined	H'FF46002C	32
	DMA reload source address register 2	DMRSADR2	R/W	Undefined	H'FF460220	32
	DMA reload destination address register 2	DMRDADR2	R/W	Undefined	H'FF460224	32
	DMA reload byte count register 2	DMRBCT2	R/W	Undefined	H'FF460228	32
	DMA control register A2	DMCNTA2	R/W	H'00000000	H'FF460410	8, 16, 32
	DMA control register B2	DMCNTB2	R/W	H'00000000	H'FF460414	8, 16, 32
3	DMA current source address register 3	DMCSADR3	R/W	Undefined	H'FF460030	32
	DMA current destination address register 3	DMCDADR3	R/W	Undefined	H'FF460034	32
	DMA current byte count register 3	DMCBCT3	R/W	Undefined	H'FF460038	32
	DMA mode register 3	DMMOD3	R/W	Undefined	H'FF46003C	32
	DMA reload source address register 3	DMRSADR3	R/W	Undefined	H'FF460230	32
	DMA reload destination address register 3	DMRDADR3	R/W	Undefined	H'FF460234	32
	DMA reload byte count register 3	DMRBCT3	R/W	Undefined	H'FF460238	32
	DMA control register A3	DMCNTA3	R/W	H'00000000	H'FF460418	8, 16, 32
	DMA control register B3	DMCNTB3	R/W	H'00000000	H'FF46041C	8, 16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	DMA current source address register 4	DMCSADR4	R/W	Undefined	H'FF460040	32
	DMA current destination address register 4	DMCDADR4	R/W	Undefined	H'FF460044	32
	DMA current byte count register 4	DMCBCT4	R/W	Undefined	H'FF460048	32
	DMA mode register 4	DMMOD4	R/W	Undefined	H'FF46004C	32
	DMA reload source address register 4	DMRSADR4	R/W	Undefined	H'FF460240	32
	DMA reload destination address register 4	DMRDADR4	R/W	Undefined	H'FF460244	32
	DMA reload byte count register 4	DMRBCT4	R/W	Undefined	H'FF460248	32
	DMA control register A4	DMCNTA4	R/W	H'00000000	H'FF460420	8, 16, 32
	DMA control register B4	DMCNTB4	R/W	H'00000000	H'FF460424	8, 16, 32
5	DMA current source address register 5	DMCSADR5	R/W	Undefined	H'FF460050	32
	DMA current destination address register 5	DMCDADR5	R/W	Undefined	H'FF460054	32
	DMA current byte count register 5	DMCBCT5	R/W	Undefined	H'FF460058	32
	DMA mode register 5	DMMOD5	R/W	Undefined	H'FF46005C	32
	DMA reload source address register 5	DMRSADR5	R/W	Undefined	H'FF460250	32
	DMA reload destination address register 5	DMRDADR5	R/W	Undefined	H'FF460254	32
	DMA reload byte count register 5	DMRBCT5	R/W	Undefined	H'FF460258	32
	DMA control register A5	DMCNTA5	R/W	H'00000000	H'FF460428	8, 16, 32
	DMA control register B5	DMCNTB5	R/W	H'00000000	H'FF46042C	8, 16, 32

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
6	DMA current source address register 6	DMCSADR6	R/W	Undefined	H'FF460060	32
	DMA current destination address register 6	DMCDADR6	R/W	Undefined	H'FF460064	32
	DMA current byte count register 6	DMCBCT6	R/W	Undefined	H'FF460068	32
	DMA mode register 6	DMMOD6	R/W	Undefined	H'FF46006C	32
	DMA reload source address register 6	DMRSADR6	R/W	Undefined	H'FF460260	32
	DMA reload destination address register 6	DMRDADR6	R/W	Undefined	H'FF460264	32
	DMA reload byte count register 6	DMRBCT6	R/W	Undefined	H'FF460268	32
	DMA control register A6	DMCNTA6	R/W	H'00000000	H'FF460430	8, 16, 32
	DMA control register B6	DMCNTB6	R/W	H'00000000	H'FF460434	8, 16, 32
7	DMA current source address register 7	DMCSADR7	R/W	Undefined	H'FF460070	32
	DMA current destination address register 7	DMCDADR7	R/W	Undefined	H'FF460074	32
	DMA current byte count register 7	DMCBCT7	R/W	Undefined	H'FF460078	32
	DMA mode register 7	DMMOD7	R/W	Undefined	H'FF46007C	32
	DMA reload source address register 7	DMRSADR7	R/W	Undefined	H'FF460270	32
	DMA reload destination address register 7	DMRDADR7	R/W	Undefined	H'FF460274	32
	DMA reload byte count register 7	DMRBCT7	R/W	Undefined	H'FF460278	32
	DMA control register A7	DMCNTA7	R/W	H'00000000	H'FF460438	8, 16, 32
	DMA control register B7	DMCNTB7	R/W	H'00000000	H'FF46043C	8, 16, 32
Common	DMA activation control register	DMSCNT	R/W	H'00000000	H'FF460500	8, 16, 32
	DMA interrupt control register	DMICNT	R/W	H'00000000	H'FF460508	8, 16, 32
	DMA common interrupt control register	DMICNTA	RW	H'00000000	H'FF46050C	8, 16, 32
	DMA interrupt status register	DMISTS	R	H'00000000	H'FF460510	8, 16, 32
	DMA transfer end detection register	DMEDET	R/W	H'00000000	H'FF460514	8, 16, 32
	DMA arbitration status register	DMASTS	R/W	H'00000000	H'FF460518	8, 16, 32

11.3.1 DMA Current Source Address Register (DMCSADR)

DMCSADR is a register used to specify the start address of the transfer source.

The value in this register is transferred to the working source-address register at the start of DMA transfer.

The default behavior is for the contents of the working source-address register to be returned on completion of single operand transfer. However, the contents of the working source address register are not returned in two cases: when the rotate setting (SAMOD = 011) is made for the source address and when the source-address reload function is enabled. In the latter case, the contents of the DMA reload source address register (DMRSADRn) are returned to this register on completion of DMA transfer.

This register must be set before transfer is initiated, regardless of whether the reload function is enabled or disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSA															
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
	CSA															
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	CSA	Undefined	R/W	Holds source address bits A31 to A0

Notes: 1. Set this register so that DMA transfer is performed within the correctly aligned address boundaries for the transfer sizes listed below.
- When the transfer size is set to 16 bits (SZSEL = "001"): (b0) = "0".
- When the transfer size is set to 32 bits (SZSEL = "010"): (b1, b0) = (0, 0).

2. Only write to this register when single operand transfer is not in process on the corresponding channel (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is "0") and DMA transfer is disabled (DMST in the DMA activation control register (DMSCNT) or DEN in DMA control register B for the channel (DMCNTBn) is set to "0"). Operation is not guaranteed if this register is written to when both conditions are not satisfied.

11.3.2 DMA Current Destination Address Register (DMCDADR)

DMCDADR is a register used to specify the start address of the transfer destination.

The value in this register is transferred to the working destination-address register at the start of DMA transfer.

The default behavior is for the contents of the working destination-address register to be returned on completion of each single operand transfer. However, the contents of the working destination-address register are not returned in two cases: when the rotate setting (SAMOD = 011) is made for the destination address and when the destination-address reload function is enabled. In the latter case, the contents of the DMA reload destination address register (DMRDADRn) are returned to this register on completion of DMA transfer.

This register must be set before transfer is initiated, regardless of whether the reload function is enabled or disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDA															
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
	CDA															
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	CDA	Undefined	R/W	Holds destination address bits A31 to A0

Notes: 1. Set this register so that DMA transfer is performed within the correctly aligned address boundaries for the transfer sizes listed below.
- When the transfer size is set to 16 bits (SZSEL = "001"): (b0) = "0".
- When the transfer size is set to 32 bits (SZSEL = "010"): (b1, b0) = (0, 0).

2. Only write to this register when single operand transfer is not in process on the corresponding channel (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is "0") and DMA transfer is disabled (DMST in the DMA activation control register (DMSCNT) or DEN in DMA control register B for the channel (DMCNTBn) is set to "0"). Operation is not guaranteed if this register is written to when both conditions are not satisfied.

11.3.3 DMA Current Byte Count Register (DMCBCT)

DMCBCT is a register used to specify the number of bytes to be transferred by DMA.

The value in this register is transferred to the working byte-count register at the start of DMA transfer, and is then decremented by the number of bytes transferred on each unit data transfer. Decrementation is thus by the following values.

- When the transfer size is set to 8 bits (SZSEL = "000"): –1
- When the transfer size is set to 16 bits (SZSEL = "001"): –2
- When the transfer size is set to 32 bits (SZSEL = "010"): –4

When the value in the working byte count register reaches H'000 0000, DMA transfer ends (an end due to byte count "0"). The corresponding bit of the DMA transfer end detection register (DMEDET) is set to 1.

If the byte count reload function is disabled, the contents of the working byte count register are returned to this register at the moment the channel for DMA transfer switches or DMA transfer ends. If the byte count reload function is enabled, the contents of the DMA reload byte counter register (DMRBCTn) are returned to this register.

This register must be set before transfer is initiated, regardless of whether the reload function is enabled or disabled.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CBC									
Initial value:	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CBC															
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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	CBC	Undefined	R/W	Number of bytes to be DMA-transferred.

Notes: 1. Note that a setting of H'000 0000 leads to transfer of the maximum number of bytes, i.e. 64 Mbytes.

2. Set this register so that DMA transfer is performed within the correctly aligned address boundaries for the transfer sizes listed below.
   - When the transfer size is set to 16 bits (SZSEL = "001"): (b0) = "0".
   - When the transfer size is set to 32 bits (SZSEL = "010"): (b1, b0) = (0, 0).
3. Only write to this register when single operand transfer is not in process on the corresponding channel (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is "0") and DMA transfer is disabled (DMST in the DMA activation control register (DMSCNT) or DEN in DMA control register B for the channel (DMCNTBn) is set to "0"). Operation is not guaranteed if this register is written to when both conditions are not satisfied.

11.3.4 DMA Reload Source Address Register (DMRSADR)

DMRSADR is used to set an address for reloading to the DMA current source address register (DMCSADRn).

To enable reloading, set the DMA source address reload function enable bit (SRLOD) in DMA control register A (DMCNTAn) for the channel to "1". In this case, set both the DMA current source address register (DMCSADRn) and DMA reload source address register (DMRSADRn).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSA															
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
	RSA															
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	RSA	Undefined	R/W	Holds source address bits A31 to A0 for reloading

Note: Set this register so that DMA transfer is performed within the correctly aligned address boundaries for the transfer sizes listed below.
- When the transfer size is set to 16 bits (SZSEL = "001"): (b0) = "0".
- When the transfer size is set to 32 bits (SZSEL = "010"): (b1, b0) = (0, 0).

11.3.5 DMA Reload Destination Address Register (DMRDADR)

DMRDADR is a register used to set an address for reloading to the DMA current destination address register (DMCDADRn).

To enable reloading, set the DMA destination address reload function enable bit (DRLOD) in DMA control register A (DMCNTAn) to 1. In this case, set both the DMA current destination address register (DMCDADRn) and DMA reload destination address register (DMRDADRn).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDA															
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
	RDA															
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	RDA	Undefined	R/W	Holds destination address bits A31 to A0 for reloading

Note: Set this register so that DMA transfer is performed within the correctly aligned address boundaries for the transfer sizes listed below.

- When the transfer size is set to 16 bits (SZSEL = "001"): (b0) = "0".
- When the transfer size is set to 32 bits (SZSEL = "010"): (b1, b0) = (0, 0).

11.3.6 DMA Reload Byte Count Register (DMRBCT)

DMRBCT is a register used to set the number for reloading to the DMA current byte count register (DMCBCTn).

To enable reloading, set the DMA byte count reload function enable bit (BRLOD) in the DMA control register A (DMCNTAn) to 1. In this case, set both the DMA current byte count register (DMCBTn) and DMA reload byte count address register (DMRBCTn).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	RBC									
Initial value:	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBC															
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 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 0	RBC	Undefined	R/W	Number of bytes to be DMA-transferred after reloading

Note: Set this register so that DMA transfer is performed within the correctly aligned address boundaries for the transfer sizes listed below.

- When the transfer size is set to 16 bits (SZSEL = "001"): (b0) = "0".
- When the transfer size is set to 32 bits (SZSEL = "010"): (b1, b0) = (0, 0).

11.3.7 DMA Mode Register (DMMOD)

DMMOD controls the amount of data, data size selection, address direction, and various types of signal outputs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	OPSEL[3:0]				—	—	—	—	—	SZSEL[2:0]		
Initial value:	0	0	0	0	—	—	—	—	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SAMOD[2:0]			—	DAMOD[2:0]			—	—	—	—	SACT	DACT	DTCM[1:0]	
Initial value:	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/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	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
27 to 24	OPSEL [3:0]	Undefined	R/W	Number of Data Transfers in Single Operand Transfer Selection These bits are used to specify the number of single data transfers in single operand transfer. The amount of data specified by this bit is transferred continuously. Channel arbitration is not executed until this amount of data has been transferred (single operand transfer). These bits are invalid when non-stop transfer (DSEL = "11") is specified in the DMA transfer condition selection bits (DSEL) of DMA control register A (DMCNTAn). Note: Set the DMA current byte count register (DMCBCTn) so that DMCBCTn becomes H'000 0000 on transfer of the last data of the operand transfer. • When the transfer size is set to 8 bits (SZSEL = "000"): Integer multiple of the number of data transferred in each single operand transfer (× 1, × 2, × 3, and so on) • When the transfer size is set to 16 bits (SZSEL = "001"): one operand transfer data number multiplied by two (× 2, × 4, × 6, and so on) • When the transfer size is set to 32 bits (SZSEL = "010"): one operand transfer data number multiplied by four (× 4, × 8, × 12, and so on) Operation is not guaranteed when values other than the above are set. For details, see section 11.3.3, DMA Current Byte Count Register (DMCBCT) and section 11.3.6, DMA Reload Byte Count Register (DMRBCT).) 0000: 1 datum 0001: 2 data 0010: 4 data 0011: 8 data 0100: 16 data 0101: 32 data 0110: 64 data 0111: 128 data 1000 to 1111: Setting prohibited
23 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	SZSEL[2:0]	Undefined	R/W	Transfer Data Size Selection These bits are used to specify the number of bits transferred in each single data transfer. The unit for transfer can be selected as byte (8 bit), word (16 bit), or longword (32 bit). For details, see section 11.9, Units of Transfer and Positioning of Bytes for Transfer. Set the transfer size so that it doesn't exceed the widths of the data buses supported by the source and destination for DMA transfer. The bus widths of the data buses are fixed by hardware. 000: Byte (8 bits) 001: Word (16 bits) 010: Longword (32 bits) 011 to 111: Setting prohibited
15	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
14 to 12	SAMOD [2:0]	Undefined	R/W	Source Address Direction Control These bits are used to specify the direction of counting for the source address. 000: Fixed 001: Incrementation 010: Decrementation 011: Rotation 100 to 111: Setting prohibited
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	DAMOD [2:0]	Undefined	R/W	Destination Address Direction Control These bits are used to specify the direction of counting for the source address. 000: Fixed 001: Incrementation 010: Decrementation 011: Rotation 100 to 111: Setting prohibited

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	SACT	Undefined	R/W	DMA Active Signal Output for Source This bit is used to control the output of the DMA-active signal (DACT) for the source corresponding to the requesting source setting in the DCTG bits. When this bit is set to "0", output of the DACT signal is disabled and the signal is fixed high unless the level changes because of the DACT bit setting. When this bit is set to "1", output of the DACT signal is valid ("L") from the next cycle after the start of the DMAC read cycle. However, while output of the DACT signal is enabled when the DMA request source selection bits (DCTG) are set for software triggering, a valid DACT signal cannot be output when the requesting source is an on-chip peripheral circuit (DCTG), regardless of the setting of the SACT bits. 0: Stops output of the DMA-active signal for the source 1: Selects output of the DMA-active signal for the source during read access

Bit	Bit Name	Initial Value	R/W	Description
2	DACT	Undefined	R/W	DMA Active Signal Output for Destination This bit is used to control the output of the DMA-active signal (DACT) for the destination corresponding to the request source setting in the DCTG bits. When this bit is set to "0", output of the DACT signal is disabled and fixed high unless the level changes because of the SACT bit setting. When this bit is set to "1", output of the DACT signal is valid ("L") from the next cycle after the start of the DMAC read cycle. However, while output of the DACT signal is enabled when the DMA request source selection (DCTG) bits are set for software triggering, a valid DACT signal cannot be output when the requesting source is an on-chip peripheral circuit (DCTG), regardless of the setting of the DACT bit. 0: Stops output of the DMA-active signal for the destination 1: Selects output of the DMA-active signal for the destination during write access

Bit	Bit Name	Initial Value	R/W	Description
1, 0	DTCM[1:0]	Undefined	R/W	DMA End Signal Output Control These bits are used to control the output of the DMA end signal (DTEND) when the DMA transfer end condition is detected. When the bits are set to "00", DTEND signals on completion of DMA transfer are disabled and the DTEND line is fixed high. When these bits are set to "10", the DTEND signal goes low (is active) in the cycle after the read cycle immediately preceding completion of DMA transfer. When these bits are set to "10", the DTEND signal is active in the cycle after the write cycle immediately preceding completion of DMA transfer. When these bits are set to "11", the DTEND signal is active for the period of one clock cycle at the same time as the DMA transfer end interrupt (for details, see figure 11.9.) However, while output of the DTEND signal is enabled when the DMA request source selection bits (DCTG) are set for software triggering, a valid DTEND signal cannot be output when the requesting source is an on-chip peripheral circuit (DCTG), regardless of the setting of the DTEND bits. 00: Stops output of the DTEND signal 01: The DTEND signal is output on the last read cycle 10: The DTEND signal is output on the last write cycle 11: The DTEND signal is output after DMA has been completed

Note: Only write to this register when the corresponding channel is not engaged in single operand transfer (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is "0") and DMA transfer is disabled (DMST in the DMA activation control register (DMSCNT) or DEN in DMA control register B for the channel (DMCNTBn) is set to "0"). Operation is not guaranteed if this register is written to when both conditions are not satisfied.
When SACT and DACT are set to 1, output of a low DACT signal from the cycle following a DMAC read or write cycle is enabled.

Table 11.3 shows the DMA source/destination address registers. For details on the rotation address "indexing" mode, see section 11.11, Rotate Function. Note that when performing pipelined transfer to or from external devices and modules that support burst access, make sure to set the direction bits to select address incrementation ("001") or rotation ("011").

Table 11.3 Increment/Decrement for DMA Source/Destination Address Registers

Transfer data size selection bits SZSEL	Address Indexing Mode SAMOD or DAMOD			
	"000" (fixed)	"001" (plus direction)	"010" (minus direction)	"011" (rotation)
"000" (8 bits)	±0	+1	–1	+1
"001" (16 bits)	±0	+2	–2	+2
"010" (32 bits)	±0	+4	–4	+4

11.3.8 DMA Control Register A (DMCNTA)

DMCNTA handles the selections of the transfer mode and the condition of transfer, control of reload functions, and selection of DMA sources.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MDSEL[1:0]		—	—	DSEL[1:0]		—	—	—	—	—	—	STRG[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	R	R/W	R/W	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
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	—	—	DCTG[5:0]					
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	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	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
29, 28	MDSEL [1:0]	00	R/W	DMA Transfer Mode Selection These bits are used to specify the DMA transfer mode. Setting these bits to "00" selects cycle-stealing transfer mode. Setting these bits to "01" selects pipelined transfer mode. Do not set these bits to "10" or "11". Operation is not guaranteed if these settings are made. For details, see section 11.4.1, DMA Transfer Mode. 00: Cycle-stealing transfer 01: Pipelined transfer 10: Setting prohibited 11: Setting prohibited Note: Pipelined transfer through a single BIU is not possible. For details on the BIU, see section 11.1, Features.
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25, 24	DSEL[1:0]	00	R/W	DMA Transfer Condition Selection These bits are used to specify the conditions of DMA transfer. Setting these bits to "00" selects single operand transfer. Setting these bits to "01" selects sequential operand transfer. Setting these bits to "11" selects non-stop transfer. For details, see section 11.4.2, DMA Transfer Condition. Do not set these bits to "10". Operation is not guaranteed if this setting is made. 00: Unit operand transfer 01: Sequential operand transfer 10: Setting prohibited 11: Non-stop transfer
23 to 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	STRG[1:0]	00	R/W	Input Sense Mode Selection These bits specify input sense modes for DMA request signals input to the DMAC. The requesting source is that selected from among the possible sources by the DMA request source selection bits (DCTG). Select rising edge sense by setting these bits to "00" if the software trigger (DCTG = "000000") and pins DREQ0 to DREQ3 are selected as the source for DMA requests. Select falling edge sense by setting the bits to "10" when operation is with IIC3, SCIF, SSI, RCAN-ET, MTU2, or ADC (DCTG = "000101" to "100100"). Table 11.4 shows the relationships between DMA request sources and the possible input sense modes. 00: Rising edge 01: High level 10: Falling edge 11: Low level
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10	BRLOD	0	R/W	DMA Byte Count Reload Function Enable This bit specifies whether to reload the byte counter or not when the DMA transfer end condition is detected. When this bit is cleared to "0", no reload is executed. When this bit is set to "1" and the DMA transfer end condition is detected, the DMA current byte counter register (DMCBCTn) is reloaded with the value in the DMA reload byte count register (DMRBCTn). 0: Byte count reload function disabled 1: Byte count reload function enabled

Bit	Bit Name	Initial Value	R/W	Description
9	SRLOD	0	R/W	DMA Source Address Reload Function Enable This bit specifies whether or not the source address is reloaded when the DMA transfer end condition is detected. When this bit is cleared to "0", reloading is not executed. When this bit is set to "1" and the DMA transfer end condition is detected, the DMA current source address register (DMCSADRn) is reloaded with the value of the DMA reload source address register (DMRSADRn). 0: Source address reload function disabled 1: Source address reload function enabled
8	DRLOD	0	R/W	DMA Destination Address Reload Function Enable This bit specifies whether or not the destination address is reloaded when the DMA transfer end condition is detected. When this bit is cleared to "0", reloading is not re-executed. When this bit is set to "1" and the DMA transfer end condition is detected, the DMA current destination address register (DMCDADRn) is reloaded with the value of the DMA reload destination address register (DMRDADRn). 0: Destination address reload function disabled 1: Destination address reload function enabled
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	DCTG[5:0]	000000	R/W	DMA Request Source Selection These bits specify the source of DMA requests. When selecting IIC3, SCIF, RCAN-ET, MTU2, or ADC as the source, set the DMA transfer request enable bits in DREQER0 to DREQER3 of the interrupt controller. For the settings of DREQER0–3, see section 6, Interrupt Controller (INTC). 000000: Software trigger 000001: DREQ0 pin 000010: DREQ1 pin 000011: DREQ2 pin 000100: DREQ3 pin 000101: IIC3 0ch RX 000110: IIC3 0ch TX 000111: IIC3 1ch RX 001000: IIC3 1ch TX 001001: IIC3 2ch RX 001010: IIC3 2ch TX 001011: SCIF 0ch RX 001100: SCIF 0ch TX 001101: SCIF 1ch RX 001110: SCIF 1ch TX 001111: SCIF 2ch RX 010000: SCIF 2ch TX 010001: SCIF 3ch RX 010010: SCIF 3ch TX 010011: SCIF 4ch RX 010100: SCIF 4ch TX 010101: SCIF 5ch RX 010110: SCIF 5ch TX 010111: SCIF 6ch RX 011000: SCIF 6ch TX 011001: SCIF 7ch RX 011010: SCIF 7ch TX

Bit	Bit Name	Initial Value	R/W	Description
5 to 0	DCTG[5:0]	000000	R/W	011011: SSI 0ch
				011100: SSI 1ch
				011101: RCAN-ET 0ch
				011110: RCAN-ET 1ch
				011111: MTU2 0ch
				100000: MTU2 1ch
				100001: MTU2 2ch
				100010: MTU2 3ch
				100011: MTU2 4ch
				100100: ADC
				100101 to 111111: Setting prohibited

Note: Only write to bits of this register other than the reload function enable bits (BRLOD, SRLOD, and DRLOD) when a transfer operation is not in process on the corresponding channel (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is "0") and DMA transfer is disabled (DMST in the DMA activation control register (DMSCNT) or DEN in DMA control register B for the channel (DMCNTBn) is set to "0"). Operation is not guaranteed if this register is written to when both conditions are not satisfied.

Table 11.4 Relationships between DMA Request Sources and Input Sense Mode

	STRG Bit Settings				
DMA Request Source	**00: Rising Edge Sense**	**01: High Level Sense**	**10: Falling Edge Sense**	**11: Low Level Sense**	**DCTG Bit Setting**
Software trigger	√	×	×	×	000000
DREQ0 pin	√	√	√	√	000001
DREQ1 pin	√	√	√	√	000010
DREQ2 pin	√	√	√	√	000011
DREQ3 pin	√	√	√	√	000100
IIC3 0ch RX	×	×	√	×	000101
IIC3 0ch TX	×	×	√	×	000110
IIC3 1ch RX	×	×	√	×	000111
IIC3 1ch TX	×	×	√	×	001000
IIC3 2ch RX	×	×	√	×	001001
IIC3 2ch TX	×	×	√	×	001010
SCIF 0ch RX	×	×	√	×	001011
SCIF 0ch TX	×	×	√	×	001100
SCIF 1ch RX	×	×	√	×	001101
SCIF 1ch TX	×	×	√	×	001110
SCIF 2ch RX	×	×	√	×	001111
SCIF 2ch TX	×	×	√	×	010000
SCIF 3ch RX	×	×	√	×	010001
SCIF 3ch TX	×	×	√	×	010010
SCIF 4ch RX	×	×	√	×	010011
SCIF 4ch TX	×	×	√	×	010100
SCIF 5ch RX	×	×	√	×	010101
SCIF 5ch TX	×	×	√	×	010110
SCIF 6ch RX	×	×	√	×	010111
SCIF 6ch TX	×	×	√	×	011000
SCIF 7ch RX	×	×	√	×	011001
SCIF 7ch TX	×	×	√	×	011010

	STRG Bit Settings				
DMA Request Source	**00: Rising Edge Sense**	**01: High Level Sense**	**10: Falling Edge Sense**	**11: Low Level Sense**	**DCTG Bit Setting**
SSI 0ch	×	×	√	×	011011
SSI 1ch	×	×	√	×	011100
RCAN-ET 0ch	×	×	√	×	011101
RCAN-ET 1ch	×	×	√	×	011110
MTU2 0ch	×	×	√	×	011111
MTU2 1ch	×	×	√	×	100000
MTU2 2ch	×	×	√	×	100001
MTU2 3ch	×	×	√	×	100010
MTU2 4ch	×	×	√	×	100011
ADC	×	×	√	×	100100

[Legend]

×: Setting prohibited

√: Can be set

11.3.9 DMA Control Register B (DMCNTB)

DMCNTB enables or disables DMA transfer, clears the DMA transfer enable bit, and also clears the internal state. In addition, this register can check the status of a DMA request.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	DEN	—	—	—	—	—	—	—	DREQ
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/W	R	R	R	R	R	R	R	R/W

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	DEN	0	R/W	DMA Transfer Enable This bit is used to enable or disable DMA transfer on the corresponding channel. Clearing this bit to "0" disables DMA transfer. Setting this bit to "1" enables DMA transfer. For the activation of DMA transfer, see section 11.4.3, DMA Activation. Even when this bit is clear, the input of a DMA request to the DMAC can change the value of the DMA request bit (DREQ). When the DMA transfer enable clear bit (ECLR) is set to "1", this bit is automatically cleared to "0" on detection of the DMA transfer end condition. Clearing this bit to "0" during DNA transfer can be used to stop channel operation at the end of the current single operand transfer. For details, see section 11.6, Suspending, Restarting, and Stopping of DMA Transfer. 0: DMA transfer disabled 1: DMA transfer enabled

Bit	Bit Name	Initial Value	R/W	Description
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	DREQ	0	R/W	DMA Request This bit is used to check whether a DMA request is currently present. Furthermore, when the software trigger is selected (DCTG = "000000") by the DMA request source selection bits (DCTG), DMA requests operate through this bit. The value of this bit changes according to the state of DMA request input to the DMAC regardless of the settings of the DMAC module activation bit (DMST) and DMA transfer enable bit (DEN). The conditions for setting and clearing the bit are determined by the DMA request source selection bits (DCTG) and input sense mode selection bits (STRG) as described below. (a) When software triggering is selected (DCTG = "000000") by the DMA request source selection bits (DCTG). • Condition for setting to "1" This bit is set to "1" when a "1" is written to it by software, generating the DMA request. • Condition for clearing to "0" This bit is cleared to "0" by either of the below events. — Software writing a "0" to the bit — The start of the transfer operation corresponding to the bit setting

Bit	Bit Name	Initial Value	R/W	Description
16	DREQ	0	R/W	(b) When a source other than the software trigger is selected (DCTG = "000000") by the DMA request source selection bits (DCTG) and a level sense has been selected • Condition for setting to "1" This bit is set to "1" when the DMA request input level matches that specified in the input sense selection bits (STRG), i.e. when a DMA request exists. • Condition for clearing to "0" This bit is cleared to "0" when the level specified by the input sense selection bits (STRG) and the level on the DMA request input do not match, i.e. when there is no DMA request. The DMA request is not retained if it disappears before being accepted; that is, the DMA request bit (DREQ) is cleared to "0". To use the DREQ bit with a level sense, continue the DMA request level until the request has been accepted. Note: When a requesting source other than the software trigger is selected, do not write "1" to the DMA request bit (DREQ). If "1" is written to this bit, operation is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
16	DREQ	0	R/W	(c) When a source other than the software trigger is selected (DCTG = "000000") by the DMA request source selection bits (DCTG) and an edge sense has been selected • Condition for setting to "1" The DREQ bit is set to "1" when the edge specified by the input sense selection bits (STRG) is encountered, i.e. when a DMA request exists. Once this bit has been set to "1", regardless of the subsequent state of the DMA request signal, the DMA request bit (DREQ) remains set until a condition for clearing to "0" has been satisfied. • Condition for clearing to "0" This bit is cleared to "0" by either of the events listed below. — Software writing a "0" to this bit — The start of operand transfer corresponding to the bit Notes: 1. In a case where a source other than software triggering is selected, do not write "1" to the DMA request bit (DREQ). If "1" is written to this bit, operation is not guaranteed. 2. After setting the DMA request source selection bits (DCTG) and the input sense mode selection bits (STRG) in DMA control register A (DMCNTAn), be sure to clear the DMA request bit (DREQ) for the channel to "0" and enable DMA transfer (DMST = "1" and DEN = "1"). 0: No DMA request 1: DMA requested
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	ECLR	0	R/W	DMA Transfer Enable Clear This bit specifies whether or not to clear the DMA transfer enable bit (DEN) to "0" when the DMA transfer end condition is detected. When this bit is cleared to "0", the DMA transfer enable bit (DEN) is not cleared to "0" even when the DMA transfer end condition is detected. When this bit is set to "1", the DMA transfer enable bit (DEN) is cleared to "0" when the DMA transfer end condition is detected. Note: When a value is written to the DMA transfer enable clear bit for a channel handling single operand transfer, operation is not guaranteed. 0: Detection of the DMA transfer end condition does not clear the DMA transfer enable bit to 0 1: Detection of the DMA transfer end condition clears the DMA transfer enable bit to 0
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	DSCLR	0	R/W	DMA Internal State Clear Writing a "1" to this bit stops DMA transfer in the middle of a sequence of DMA transfer, suspending the remainder of the transfer and initializing the internal state of the DMAC. Writing a "1" to this bit only clears the transfer state of the DMAC internal circuit. The other registers are not initialized. Writing "0" is invalid and a "1" written to this bit is not retained. This bit is always read as "0". Note: This bit must only be written to when the corresponding channel is not in the midst of single operand transfer (DASTS in the channel corresponding to the DMA arbitration status register (DMASTS) is "0") and DMA transfer has been disabled (DMST in the DMA activation control register (DMSCNT) or DEN in DMA control register B (DMCNTBn) is set to "0"). Operation is not guaranteed when this bit is written to while these conditions do not apply. When reading: Always read as "0" When writing: 0: Invalid 1: Initializes the DMAC's internal state

Note: When the software trigger is selected as the DMA request source, the DMA request bit (DREQ) can be set to "1" regardless of the settings of the DMA transfer enable bit (DEN) and DMAC module activation bit (DMST) and whether or not a transfer operation is currently in progress. However, even if the software trigger is selected as the DMA request source, only clear the DMA request bit (DREQ) to "0" or write to the DMAC internal state clearing bit (DSCLR) when a transfer operation is not in process on the corresponding channel (the corresponding DASTS bit in the DMA arbitration status register (DMASTS) is "0") and DMA transfer has been disabled (DMST in the DMA activation control register (DMSCNT) or DEN in the DMA control register B (DMCNTBn) is set to "0"). Operation is not guaranteed if this register is written to when these conditions are not satisfied.

11.3.10 DMA Activation Control Register (DMSCNT)

DMSCNT controls the operation of the DMAC.

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

Bit:	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

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	DMST	0	R/W	DMAC Module Activation This bit is used to stop or activate the DMAC module. When this bit is cleared to "0", the DMAC module stops. When this bit is set to "1", the DMAC module is operational. For details, see section 11.4.3, DMA Activation, and section 11.6, Suspending, Restarting, and Stopping of DMA Transfer. 0: DMAC halted 1: DMAC operating
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

11.3.11 DMA Interrupt Control Register (DMICNT)

DMICNT controls DMA interrupts for the respective channels.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DINTM	All 0	R/W	DMA Interrupt Control These bits are used to control whether DMA transfer end interrupts for the respective channels should be generated for the interrupt controller. When a bit is cleared to "0", interrupt requests for the corresponding channel are not generated. When these bits are set to "1", DMA transfer end interrupts for the corresponding channel are generated for the interrupt controller. For details, see section 11.5.2, DMA Interrupt Requests. 0: Interrupt disabled 1: Interrupt enabled
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 24 correspond to channels 0 to 7, respectively (31: channel 0, 30: channel 1. …24: channel 7).

11.3.12 DMA Common Interrupt Control Register (DMICNTA)

DMICNTA determines which channels contribute to the output of a common interrupt request signal.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DINTA	All 0	R/W	DMA Common Interrupt Request Signal Control These bits are used to determine which channels contribute to the output of a common interrupt request signal. Channels for which the DINTA bit is set to "1" contribute to the output of a common interrupt request signal. Channels for which the DINTA bit is cleared to "0" do not contribute to the output of a common interrupt request signal. Only the states of channels for which the corresponding DINTA bit is set to "1" are reflected in the DMA interrupt status register (DMISTS) when a common interrupt request signal has been generated. For details, see section 11.5.2, DMA Interrupt Requests. 0: The channel does not contribute to the output of a common interrupt requests 1: The channel contributes to the output of a common interrupt request
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 24 correspond to channel 0 to 7, respectively (31: channel 0, 30: channel 1, …, 24: channel 7).

11.3.13 DMA Interrupt Status Register (DMISTS)

DMISTS consists of the DMA interrupt request status bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DISTS								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DISTS	All 0	R	DMA Interrupt Request Status These bits are used to verify the sources of common interrupt requests for the interrupt controller. • Condition for setting to "1" When the DMA common interrupt request signal control bit (DINTA) for a channel is set to "1" and the DMA transfer end condition is detected, the corresponding bit is set to "1". The setting of the DMA interrupt control bit (DINTM) does not affect this setting. • Condition for clearing to "0" A DISTS bit is cleared to "0" by clearing the corresponding DMA transfer end condition detection bit (DEDET) in the DMA transfer end detection register (DMEDET). For details, see section 11.5.2, DMA Interrupt Requests. 0: No interrupt request 1: An interrupt request exists
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Notes: 1. This register is read-only.
2. Bits 31 to 24 correspond to channels 0 to 7, respectively (31: channel 0, 30: channel 1, …, 24: channel 7).

11.3.14 DMA Transfer End Detection Register (DMEDET)

DMEDET verifies the status of DMA transfer end detection for each channel. Writing 0 to the DEDET bit is invalid and 1 written to the bit is not retained.

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

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

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DEDET	All 0	R/W	Values read: DMA Transfer End Condition Detection Values written: DMA Transfer End Condition Detection, DMA Interrupt Request Status Clear These bits are used to verify the status of DMA transfer end condition detection for each channel. Reading this register does not automatically clear the bits. Once a bit has been set to "1", the value is retained in the register as long as the bit is not cleared by software or a reset. • Condition for setting to "1" When the DMA transfer end condition is detected, these bits are set to "1". • Condition for clearing to "0" These bits are cleared to "0" by writing a "1" to the bits to be cleared. Write "0" to bits that are not to be cleared. While a bit is clear, it cannot be set to "1" by a write operation. When the DMA transfer end interrupt is in use and an interrupt request generated for a given channel starts to be handled, write a "1" to the corresponding DMA transfer end condition detection (DEDET) bit. When the DMA transfer end condition detection (DEDET) bits are cleared to "0", the DMA interrupt request status bit (DISTS) is also cleared. Values read: 0: DMA transfer end condition not detected 1: DMA transfer end condition detected Values written: 0: Invalid 1: Clears DMA transfer end condition detection and DMA interrupt request status
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 24 correspond to channels 0 to 7, respectively (31: channel 0, 30: channel 1, …, 24: channel 7).

11.3.15 DMA Arbitration Status Register (DMASTS)

DMASTS verifies the status of DMA transfer on each channel. Writing 0 to the DASTS bit is invalid and 1 written to the bit is not retained.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DASTS								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	DASTS	All 0	R	When read: DMA Arbitration Status When written: DMA Arbitration Status Clear These bits are used to verify the status of DMA transfer on each channel. • Condition for setting to "1" • The bit for a channel in which operand transfer (non-stop transfer) has started is set to "1". • Condition for clearing to "0" These bits are cleared to "0" by either of the following events. — Correct completion of single operand transfer (non-stop transfer). — A "1" is written to the bit. These bits are not cleared to "0" when DMAC operation is forcibly ended by the external DMA transfer forcible end signal. Write "1" to these bits to clear them. Note: In DMA transfer to external devices, the DMA arbitration status bit (DASTS) can be cleared before the end of external bus access (once the last data-write operation has started). When read: 0: Operand transfer not in progress 1: Operand transfer in progress When written: 0: Invalid 1: Clears DMA arbitration status
23 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: Bits 31 to 24 correspond to channels 0 to 7, respectively (31: channel 0, 30: channel 1, …, 24: channel 7)

11.4 Operation

11.4.1 DMA Transfer Mode

There are two DMA transfer modes — cycle-stealing mode and pipelined mode. These modes are selectable through the setting of the DMA transfer mode select bits (MDSEL) in DMA Control Register A (DMCNTAn).

Figure 11.2 gives examples of how bus mastership alternates between the DMAC and CPU in various DMA transfer modes.

(1) Cycle-stealing Transfer Mode

Cycle-stealing transfer mode is selected when the DMA transfer mode select bits are set to "00".

In cycle-stealing transfer mode, the DMAC leaves at least one cycle between the read and write access cycles of each single data transfer. During this interval, the CPU can access the same target BIU as the source or destination of its own operations. For details on the BIU, see section 11.1, Features.

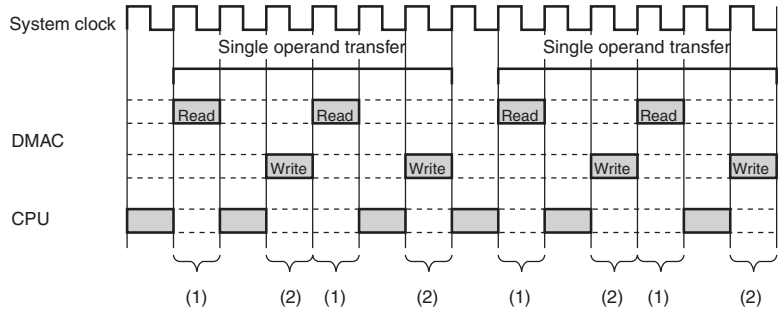
(2) Pipelined Transfer Mode

Pipelined transfer mode is selected when the DMA transfer mode select bits are set to "01".

In pipelined transfer mode, DMAC activates the bus for read or write access, or both, on consecutive cycles. Therefore, the CPU cannot access the target BIU as a source or destination during single operand transfer.

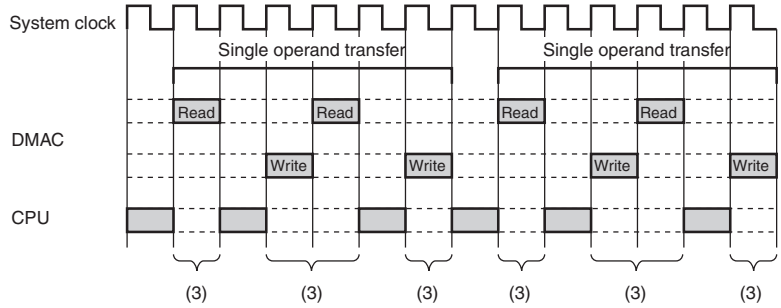
Pipelined transfer through a single BIU is not possible either.

Cycle steal transfer mode (transfer between different BIU)



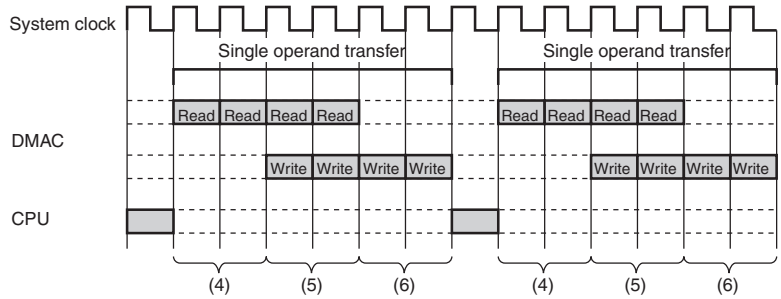
(1) CPU access to other than BIU on DMAC read side is possible
(2) CPU access to other than BIU on DMAC write side is possible

Cycle steal transfer mode (transfer in the same BIU)



(3) CPU access to other than BIU on DMAC read/write side is possible

Pipeline transfer mode (transfer between different BIU)



(4) CPU access to other than BIU on DMAC read side is possible
(5) CPU access to other than BIU on DMAC read/write side is possible
(6) CPU access to other than BIU on DMAC write side is possible

However, when a DMA access to external address space followed by CPU access to external address space is occurred, CPU access next to DMA cycle may not be occurred.

Figure 11.2 Examples of the Alternation of Bus Mastership between the DMAC and CPU in Various DMA Transfer Modes

11.4.2 DMA Transfer Condition

There are three methods of DMA transfer — the unit transfer operation, sequential operand transfer, and non-stop transfer. These are selectable through the setting of the DMA transfer condition selection bits (DSEL) in DMA Control Register A (DMCNTAn). Each of the conditions is explained below. Table 11.5 and figure 11.3 are a list and chart of the DMA transfer conditions.

(1) Unit Operand Transfer

Setting the DMA transfer condition selection bits (DSEL) to 00 selects this mode. A single DMA request initiates continuous transfer of the number of bytes selected by the OPSEL bits in the DMA mode register. If the byte counter does not reach 0 in single operand transfer, the DMA transfer is completed by repeating unit transfer operations until the byte counter does reach 0.

In the case that the DMA transfer condition is the unit operand transfer and the input sense mode of DMA request is the level sense, there is the mask period of the DMA request in the channel arbitration period after one operand transfer end (please refer to section 11.7.3, Sense Mode for DMA Requests for details). Therefore, in the channel arbitration period after one operand transfer end, in the case that there is no DMA request of the higher-priority channel than the transferring channel and there is the DMA request of the lower-priority channel than the transferring channel, the DMA transfer of the low-priority channel starts. To execute the DMA transfer of the high-priority channel in succession, please set the DMA transfer condition to the sequential operand transfer or the non-stop transfer.

(2) Sequential Operand Transfer

Setting the DMA transfer condition selection bits (DSEL) to 01 selects this mode. A single DMA request initiates transfer in units of the number of bytes selected by the OPSEL bits in the DMA mode register (i.e., unit transfer operations) until the DMA transfer is complete (i.e., until the byte counter reaches zero). Channel arbitration is performed on completion of each unit transfer operation. Transfer on the channel for the sequential operand transfer automatically resumes unless there is a DMA request from a higher-priority channel.

In the case that the DMA transfer condition is the sequential operand transfer, even if the input sense mode of DMA request is the level sense, there is no mask period before the byte count becomes 0. Therefore, the DMA transfer of the low-priority channel than the transferring channel cannot start.

(3) Non-Stop Transfer

Setting the DMA transfer condition selection bits (DSEL) to 11 selects this mode. A single DMA request initiates DMA transfer that continues until the transfer is complete (i.e., until the byte counter reaches zero). There are no gaps for channel arbitration, so even DMA requests from high-priority channels will not be accepted.

Table 11.5 List of DMA Transfer Conditions

DMA Transfer Condition Select Bits (DSEL)	DMA Transfer Condition	Remarks
DSEL = "00"	Unit operand transfer • The number of bytes selected for transfer in single operand transfer (by the OPSEL bits) is transferred in response to one DMA request. • Channel arbitration is performed on completion of each single operand transfer.	
DSEL = "01"	Sequential operand transfer • Transfer in response to a DMA request proceeds in unit transfer operations until the byte counter reaches "0". • Channel arbitration is performed on completion of each single operand transfer.	
DSEL = "11"	Non-stop transfer • Transfer in response to a DMA request proceeds continuously until the byte counter reaches "0" by one DMA request. • Once transfer has started, channel arbitration is not done until it is complete.	OPSEL bit is disabled

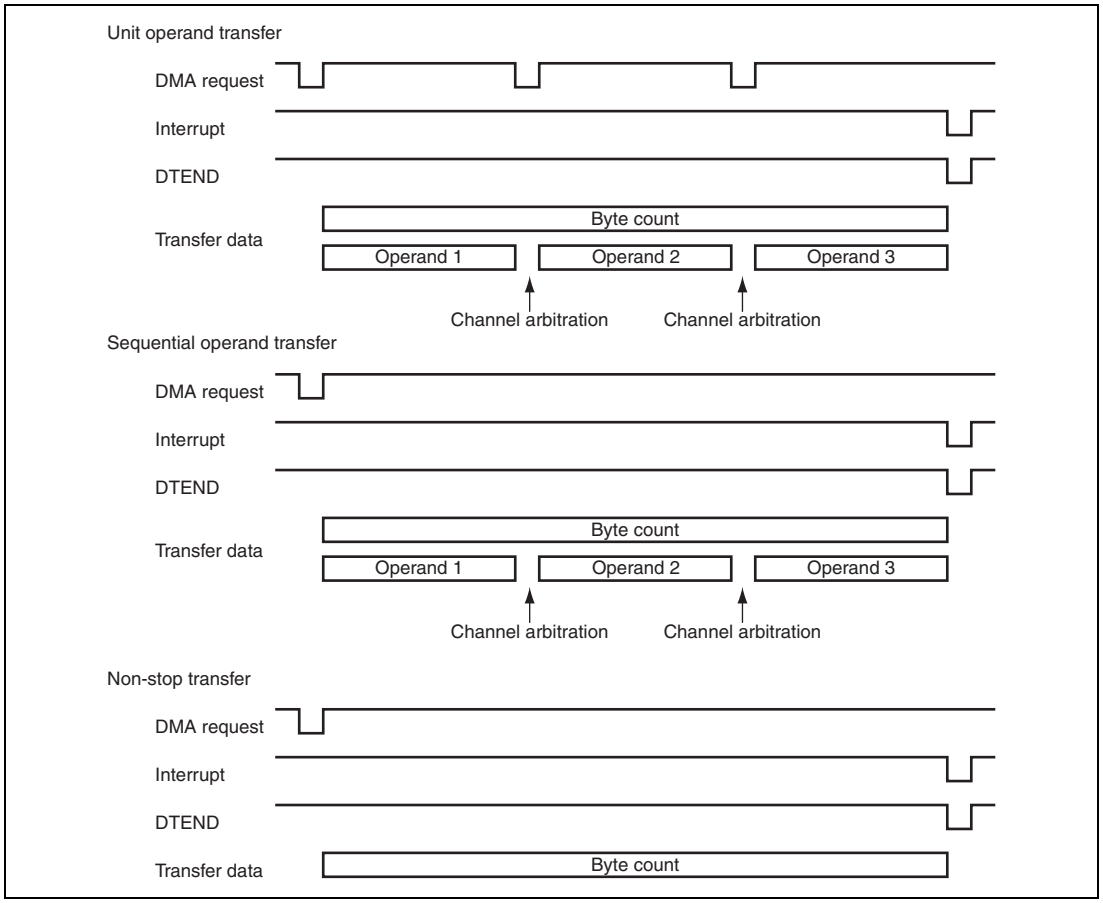


Figure 11.3 DMA Transfer Conditions

Relations between the mode and conditions of DMA transfer are shown in table 11.6.

Table 11.6 Relations between the mode and conditions of DMA transfer.

		DMA transfer mode condition		
		Unit operand transfer	**Sequential operand transfer**	**Non-stop transfer**
		DSEL = "00"	**DSEL = "01"**	**DSEL = "11"**
Transfer mode	Cycle-stealing transfer MDSEL = "00"	OK (between any two BIUs)	OK (between any two BIUs)	OK (between any two BIUs)
	Pipelined transfer MDSEL = "01"	OK (between any two BIUs)	OK (between any two BIUs)	Mainly OK* (between any two BIUs other than BIU_E)

Note: * The restriction means that non-stop transfer to the external SDRAM in pipelined transfer mode cannot be set up.

11.4.3 DMA Activation

(1) Initial Settings of the DMAC

Initial settings must be made in each of the relevant registers before the DMA transfer enable bit is set (DEN = "1"). These settings cannot be changed once transfer has started.

An example of DMAC registers that require initial settings is given below.

1. DMA mode register (DMMODn)
2. DMA control register A (DMCNTAn)
3. DMA control register B (DMCNTBn)
4. DMA current source address register (DMCSADRn)
5. DMA reload source address register (DMRSADRn) — when the reload function is used
6. DMA current destination address register (DMCDADRn)
7. DMA reload destination address register (DMRDADRn) — when the reload function is used
8. DMA current byte count register (DMCBCTn)
9. DMA reload byte count register (DMRBCTn) — when the reload function is used
10. DMA interrupt control register (DMICNT) — when an interrupt is used
11. DMA common interrupt control register (DMICNTA) — when an interrupt is used
12. DMA transfer enable bit (DEN)
13. DMA activation control register (DMSCNT)

(2) DMA Activation

DMA transfer for a channel is enabled by setting the DMA transfer enable bit (DEN) in DMA control register B for the channel and the DMAC module activation bit (DMST) in the DMAC activation register (DMSCNT) to "1".

When multiple DMA transfer requests are present, there is no complex mechanism for the determination of channel priority. The DMA request that corresponds to the highest priority channel is simply accepted and DMA transfer on that channel starts.

Whether a DMA request on a given channel is or is not present can be verified by testing the value of the DMA request bit (DREQ) in DMA control register B (DMCNTBn) for that channel.

When a DMA request is accepted and DMA transfer starts, the DMA arbitration status bit (DASTS) for the corresponding channel in the DMA arbitration status register (DMASTS) is set to "1".

11.5 Completion of DMA Transfer and Interrupts

11.5.1 Completion of DMA Transfer

When the value H'0000 0000 is transferred from the working byte count register to the DMA current byte count register (DMCBCTn) (all data has been transferred), the DMA transfer end condition is fulfilled and one DMA transfer is complete.

The operations following detection of the DMA transfer end condition are as follows.

- DMA transfer end condition

  The DMA transfer end condition detection bit (DEDET) for the corresponding channel in the DMA transfer end detection register (DMEDET) is set to "1".

- Interrupt request generation

  An interrupt request is generated for the interrupt controller according to the settings of the DMA interrupt control register (DMICNT) and the DMA common interrupt control register (DMICNTA).

- Output of DMA end signal

  The DMA end signal (DTENDm) is output according the setting of the DMA end signal output control bit (DTCM) in the DMA mode register (DMMODn) for the channel.

- Clearing the DMA transfer enable bit (DEN)

  If the DMA transfer enable clear bit (ECLR) in DMA control register B (DMCNTBn) is set to "1", the DEN bit in the DMA control register B (DMCNTBn) is cleared to "0", suspending any subsequent DMA transfer for the channel.

  If the DMA transfer enable clear bit (ECLR) is clear ("0"), the DEN bit is not cleared.

- Reloading the source address register

  If the DMA source address reload function enable bit (SRLOD) in the DMA control register A (DMCNTAn) is set to "1", the DMA current source address register (DMCSADRn) is reloaded with the value in the DMA reload source address register (DMRSADRn).

- Reloading the destination address register

  If the DMA destination address reload function enable bit (DRLOD) in DMA control register A (DMCNTAn) is set to "1", the DMA current destination address register (DMCDADRn) is reloaded with the value in the DMA reload destination address register (DMRDADRn).

- Reloading the byte count register

If the DMA byte count reload function enable bit (BRLOD) in the DMA control register A (DMCNTAn) is set to "1", the DMA current byte count register (DMCBCTn) is reloaded with the value in the DMA reload byte count register (DMRBCTn).

Note: If reloading is not to be executed, set ECLR = "1" to ensure that the DEN bit is cleared.

11.5.2 DMA Interrupt Requests

The DMAC generates two types of interrupt request signal for the interrupt controller. One consists of the interrupt request signals for the individual channels (DMINT_N) and the other is the common interrupt request signal in which the interrupt request signals from all channels are pooled to produce a common interrupt request signal (DMINTA_N).

Figure 11.4 is a block diagram showing how the per-channel and common interrupt requests are generated.

When a DMA transfer ends and the DMA interrupt control bit (DINTM) for the corresponding channel in the DMA interrupt control register (DMICNT) is set to "1", interrupt requests for the corresponding channel are generated.

Only those channels for which the DMA common interrupt request signal control bit (DINTA) in the DMA common interrupt control register (DMICNTA) is set to "1" contribute to the output of common interrupt request.

Once generated, an interrupt request is cleared to "0" by writing a "1" to the corresponding DMA transfer end condition detection bit (DEDET).

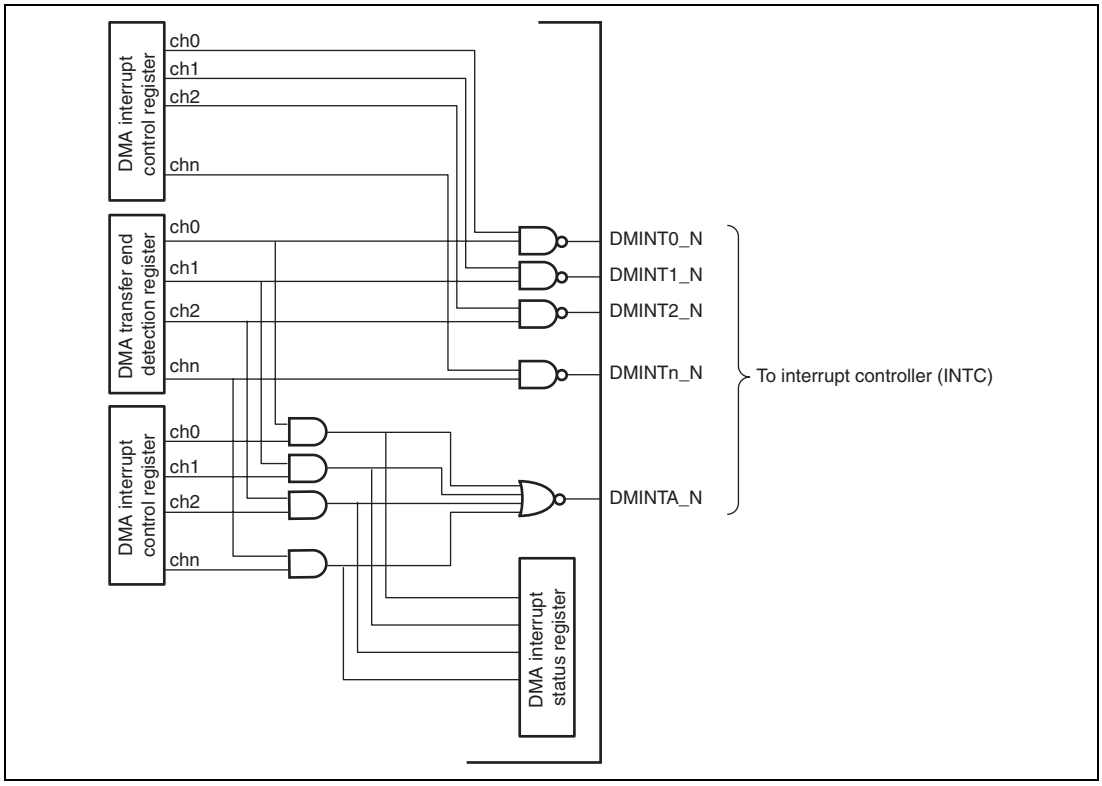


Figure 11.4 Block Diagram Showing Generation of the Per-Channel and Common Interrupt Request Signals

11.5.3 DMA End Signal Output

The form in which the DMA end signal (DTENDm) is output differs with the setting of the DMA end signal output control bit (DTCM) in the DMA mode register (DMMODn) for the corresponding channel.

- When DTCM is set to "00", output of the DTEND signal is not valid so the signal remains fixed at the "H" level when and after the DMA transfer ends.
- When DTCM is set to "01", the DTEND signal becomes active (low) one cycle after the start of the read cycle immediately before the end of DMA transfer (the read cycle for the last data transfer).
- When DTCM is set to "10", the DTEND signal becomes active for one cycle after the write cycle immediately before the end of DMA transfer (the write cycle for the last data transfer).
- When DTCM is set to "11", the DTEND signal becomes active for one clock cycle at the same time as the DMA transfer end interrupt is generated.

Output of the DTEND signal is not valid in the case of DMA requests from external peripheral circuits, so the signal remains fixed to "H" regardless of the setting of this bit.

Charts of the timing of DMA end signal output are given in figure 11.5.

Note: The BSC is provided with a write buffer. Writing data to this buffer while writing to the external devices stops bus access in the chip. Because of this, in DMA transfer to or from external devices, the DTEND signal become disabled ("H") before the end of external bus access. In this case the DTEND signal is not synchronized with the external bus access.

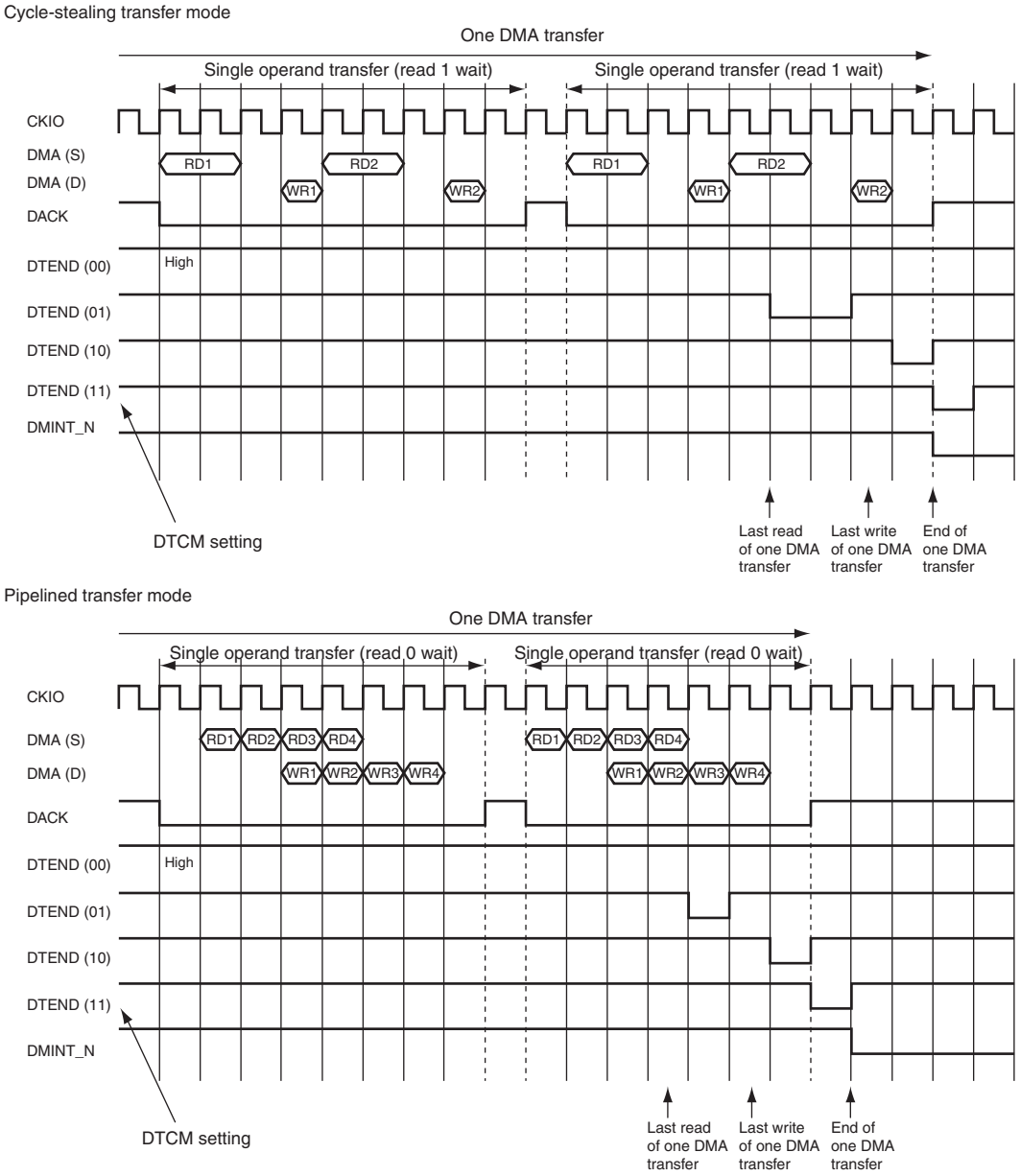


Figure 11.5 Timing of DMA End Signal Output

11.6 Suspending, Restarting, and Stopping of DMA Transfer

11.6.1 Suspending and Restarting DMA Transfer

Transfer on all channels of the DMAC can be suspended by clearing the DMST bit in the DMA activation control register (DMSCNT) to "0". Transfer on a specific channel can also be suspended by clearing the DMA transfer enable bit (DEN) in DMA control register B (DMCNTBn) for that channel.

If the DMST bit or the corresponding DEN bit is cleared to "0" while single operand transfer or sequential operand transfer is in progress, transfer is suspended on completion of the current single operand transfer regardless of the transfer mode (whether transfer is in cycle-stealing or pipelined mode).

When transfer in the non-stop transfer condition is in progress, DMA transfer is not suspended and continues to completion (until the byte counter reaches "0") even if the DMST bit or corresponding DEN bit is cleared to "0".

To restart DMA transfer on a channel for which transfer has been suspended, set (to "1") whichever of DMST and the corresponding DEN bit has been cleared.

11.6.2 Stopping DMA Transfer on Any Channel

To stop transfer on any channel, suspend transfer on that channel and then initialize the interior state of the DMAC for that channel by setting the DMAC internal state clear bit (DSCLR) in the corresponding DMA control register B (DMCNTBn). In this case, only the transfer state of the DMAC internal circuits is initialized; the registers retain their values.

11.7 DMA Requests

11.7.1 Sources of DMA Requests

The 37 sources of DMA requests include the software trigger and various DMA request signal inputs.

The DMA request source for each channel is specified by the DMA request source select bits (DTCG) in the corresponding DMA control register A (DMCNTAn).

11.7.2 Synchronous Circuits for DMA Request Signals

For each channel of the DMAC, a synchronous circuit is incorporated to manage DMA requests, which are asynchronously input. As a result, a blank period of a few clock cycles appears between activation of the DMA request and actual reflection of the request in the DMA request bits (DREQ) of DMA control register B (DMCNTBn). Figure 11.6 shows an example of timing between the input of a DMA request and the DMA request bit.

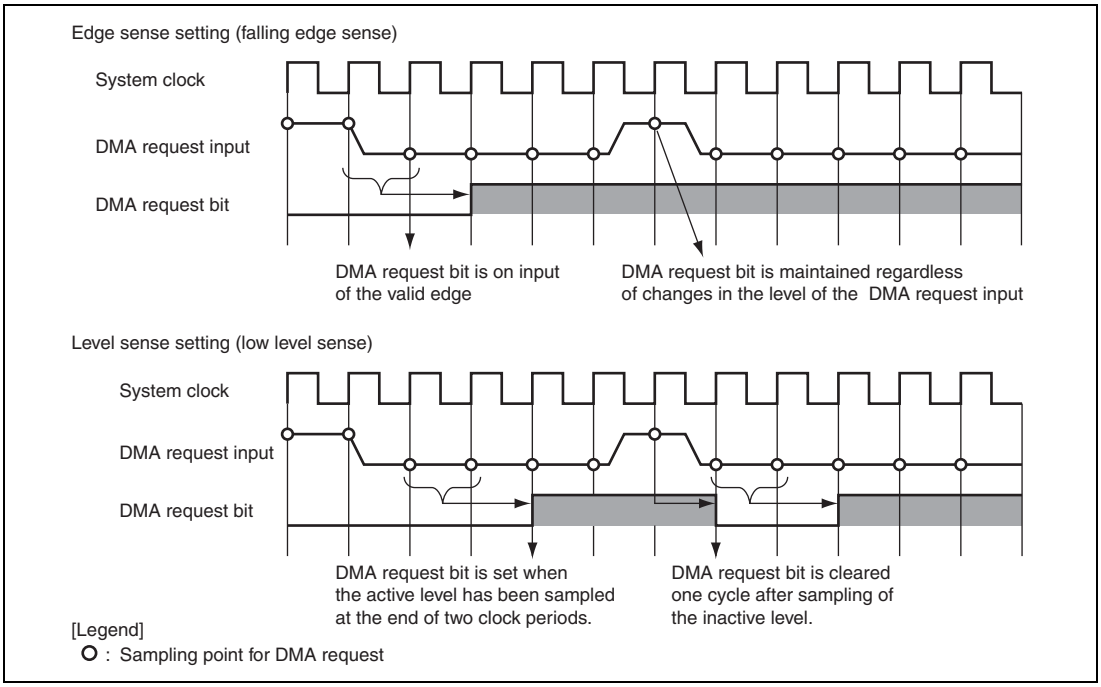


Figure 11.6 Example of Timing between DMA Request Input and DMA Request Bit

11.7.3 Sense Mode for DMA Requests

When pins DREQ0 to DREQ3 (DCTG = "000001" to "000100") are specified by the DMA request source selection bits (DTCG), either level sense or edge sense might be required. Make the appropriate setting ("01" or "11" for level sense and "00" or "10" for edge sense) in the input sense selection bits (STRG) of DMA control register A (DMCNTAn).

When the software trigger (DCTG = "000000") is selected as a DMA request source, set these bits to "00" to select the rising-edge sense. When IIC3, SCIF, SSI, RCAN-ET, MTU2, or ADC (DCTG = "000101" to "100101") is selected, set the bits to "10" to select the falling-edge sense. Table 11.4 shows the relationships between the DMA request sources and input sense mode.

Below are further details on level- and edge-sense operation.

(1) Level Sense

When a level sense is specified (STRG = "01" or "11"), one level of the DMA request signal indicates the DMA request. Since DMA requests detected in this way are not retained in the DMAC, maintain the requesting level until acceptance of the DMA request has been confirmed.

Figure 11.7 is an example of DMA request reception processing when a level sense has been selected.

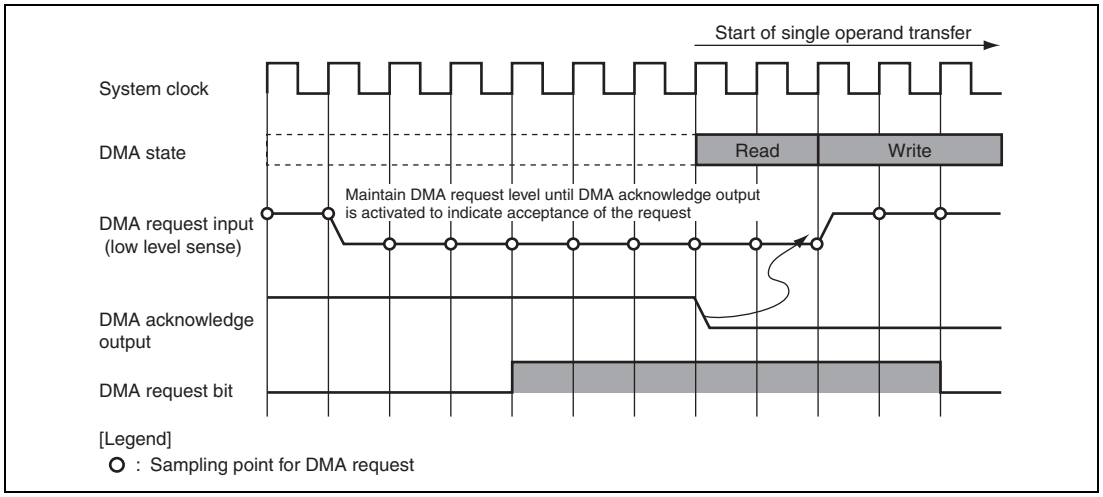


Figure 11.7 Example of DMA Request Reception Processing for a Level Sense

When a level sense has been selected, DMA request bit for the channel is masked over the period from the start of the last write access of single operand transfer until four clock pulses (system

clock) after the end of the single operand transfer. This provides a margin in which continued requests for DMA transfer on the same channel are rejected.

Figure 11.8 shows the period over which DMA request bit is masked when a level sense has been selected.

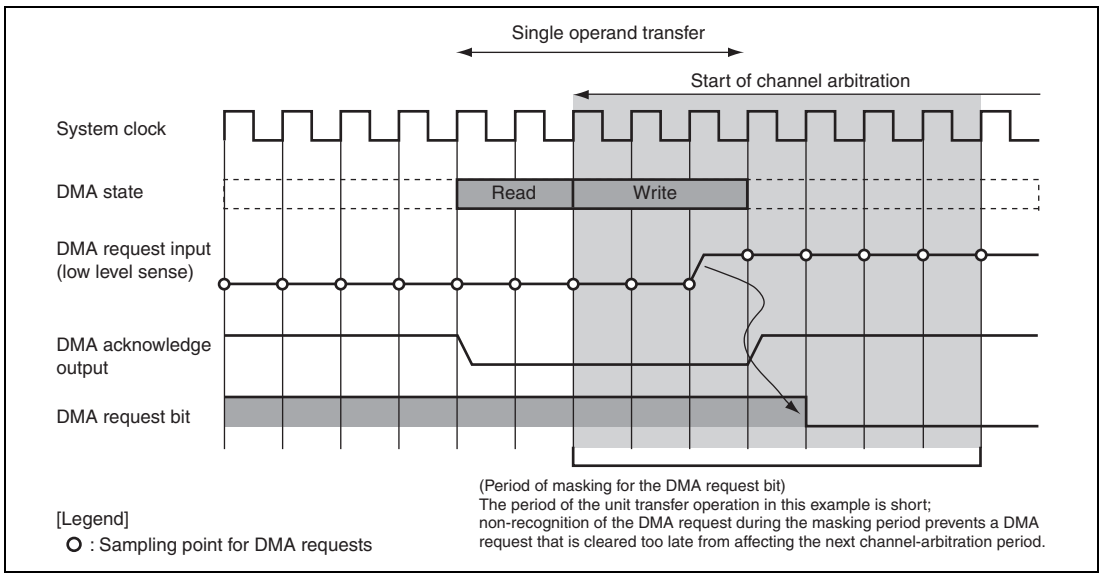


Figure 11.8 Period over which DMA Request Bit is Masked when a Level Sense is Selected

Therefore, for a channel on which level sense has been selected, even when the DMA request signal level is maintained (requesting further DMA transfer) well after the DMA request has been accepted and handled, DMA requests on other channels, if they exist, are accepted. This is because the DMA request on the channel on which level sense has been selected is not considered to exist during the DMA request bit masking period.

In the case of sequential operand transfer, masking is only applied from the end of operand transfer, i.e. when the byte count is 0. The DMA request is not masked while the byte count is non-zero, so channel arbitration is executed without masking of the DMA request during the actual unit transfer operation.

In the case of non-stop transfer, masking is only enabled from the end of the transfer operation, i.e. when the byte count is 0.

If the DMA transfer is not done sequentially, the DMA request must be canceled within three cycles after the end of single operand transfer.

(2) Edge sense

When an edge sense is specified (STRG = "00" or "10"), the rising or falling edge of the DMA request signal indicates a DMA request.

When the selected edge is detected, the DMA request bit (DREQ) in the DMA control register B (DMCNTBn) is set to "1". After that, the value in the DMA request bit (DREQ) is retained regardless of shifts in the level of the DMA request signal. After the DMA request has been accepted and the DAM acknowledge signal output, the DMA request bit (DREQ) is automatically cleared to "0".

Since DMA requests are internally retained for a channel in edge sense mode, further occurrences of the selected edge of the DMA request signal are ignored since the DMA request bit (DREQ) has already been set back to "1".

Figure 11.9 is an example of DMA request reception processing when an edge sense is selected.

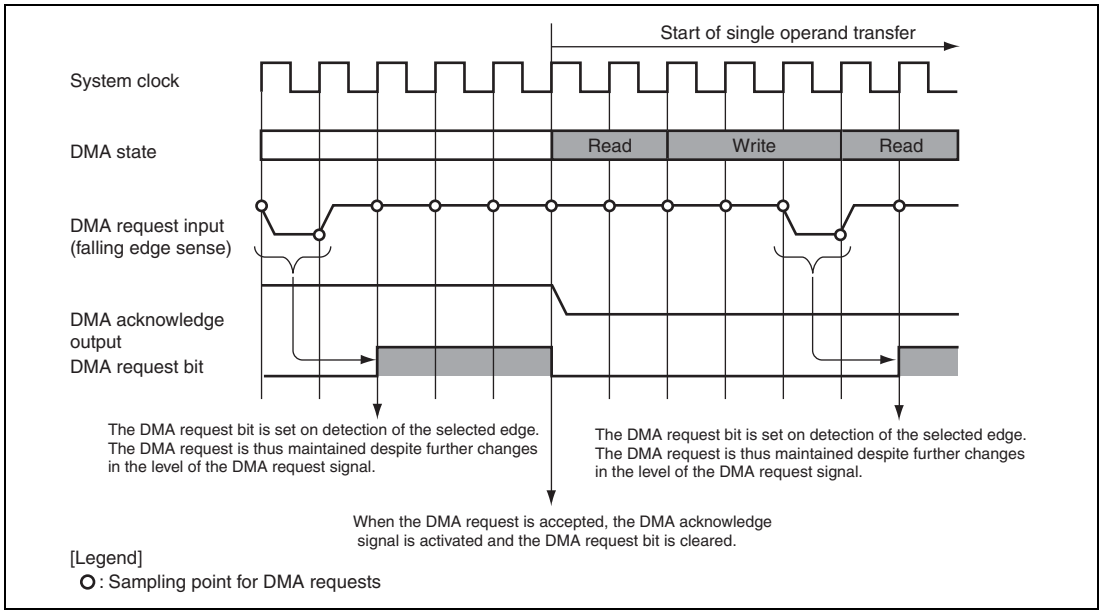


Figure 11.9 Example of DMA Request Reception Processing when an Edge Sense is Selected

11.8 Determining DMA Channel Priority

11.8.1 Channel Priority Order

Channel priority is allocated in descending order from channel 0; that is priority follows the below relation, where P indicates priority.

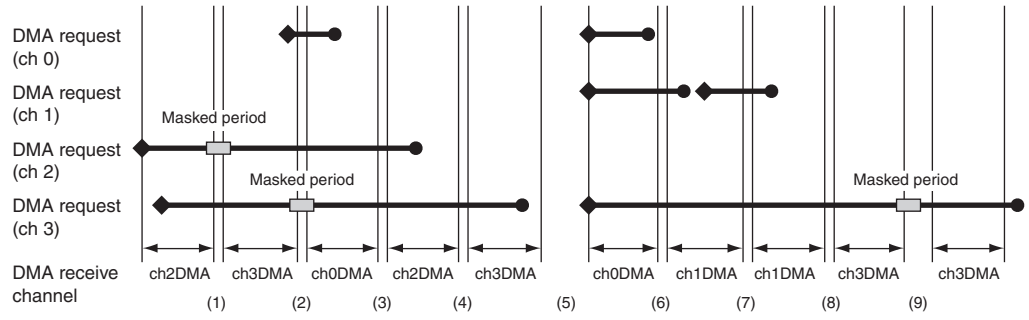
$P_{channel\ 0} > P_{channel\ 1} > P_{channel\ 3} \cdots P_{channel\ 6} > P_{channel\ 7}$. This order is fixed.

11.8.2 Operation during Multiple DMA Requests

The DMAC determines the priority every time single operand transfer is performed.

When a DMA request with a higher priority is generated during transfer for one channel, the transfer for the higher-priority channel only starts after the end of the current operand transfer. Figure 11.10 shows overall operation when multiple DMA requests are generated. The thick lines in the figure indicate the periods over which the DMA request signals are at the low level. Here channels 0, 2 and 3 are set to a level sense and channel 1 is set to an edge sense.

1. Since the channel 2 request is masked, it is regarded as non-existent. Thus, transfer on channel 3 starts up.
2. Since channel 0 has the highest priority, transfer on this channel starts up.
3. Since channel 2 has the higher priority of the requests at this point, transfer on this channel restarts.
4. Transfer on channel 3 is restarted as there are no other requests at this point.
5. When the DMA requests are simultaneously generated for channels 0, 1, and 3, transfer on channel 0 starts up because it has the highest priority.
6. After the transfer on channel 0 is complete, transfer on channel 1 starts up because it has the second highest priority.
7. A further DMA request (the selected edge) is received on channel 1 while DMA transfer is in progress. Transfer on channel 1 is thus restarted after completion of the current round of transfer on channel 1. No masking period applies in the case of edge sensing.
8. On completion of the transfer on channel 1, transfer on channel 3 starts up since there are no other requests.
9. No transfer starts up immediately after the end of the unit transfer operation on channel, since channel 3 requests are masked and there are no other requests. Transfer on channel 3 only restarts after the end of the masking period.



Notes: 1. Channels 0, 2 and 3 are set to level sensing.
2. Channel 1 is set to edge sensing.
3. Thick lines indicate periods where the corresponding DREQ bits are set.

Figure 11.10 Overall Operation during Multiple DMA Requests

11.8.3 Output of the DMA Acknowledge and DNA Active Signals

The settings of the DMA active signal output control bits for the source and destination (SACT or DACT) in the corresponding DMA mode register control the output of the DMA active signal (DACT) for a channel.

When SACT is set to 1, the DACT signal is activated in response to read access.

When DACT is set to 1, the DACT signal is activated in response to write access.

When both SACT and DACT are set to 1, the DACT signal is activated in response to read and write access.

However, DACT signals are not activated for DMA requests from external peripheral circuits, regardless of the setting of this bit.

The DMA acknowledge signal (DACK) is output throughout each single operand transfer.

Figure 11.11 is the timing chart for DMA acknowledge and DMA active signal output.

Note: The BSC is provided with a write buffer. Writing data to this buffer while writing to the external devices stops bus access in the chip. Because of this, in DMA transfer to or from external devices, the DACT or DACK signal become disabled ("H") before the end of external bus access. In this case, these signals are not synchronized with the external bus access.

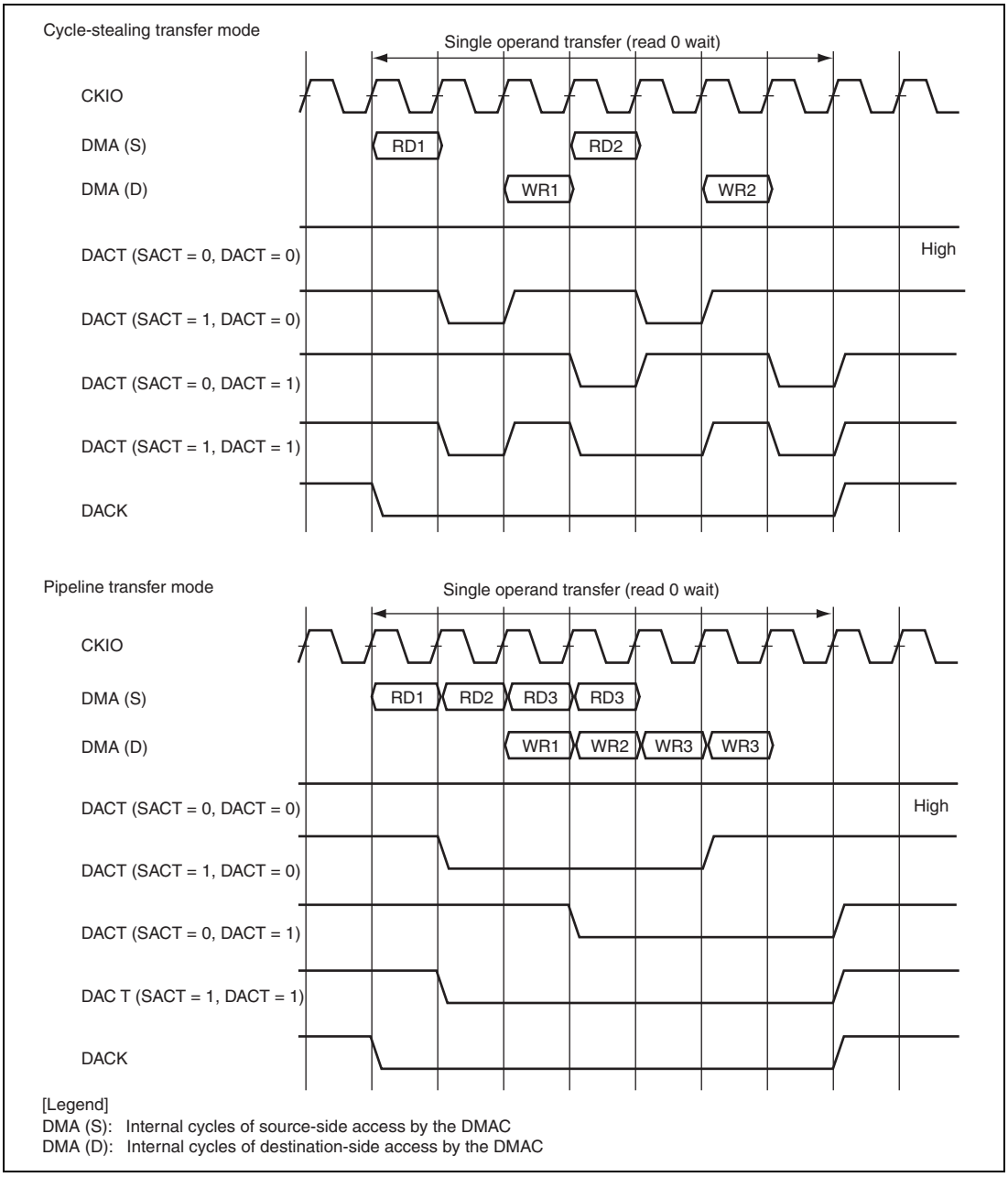


Figure 11.11 Timing of DMA Acknowledge and DNA Active Signal Output

11.9 Units of Transfer and Positioning of Bytes for Transfer

The number of bits (transfer data size) for a single data transfer can be selected from among the byte (8 bits), word (16 bits), and the longword (32 bits).

Figure 11.12 is an example of DMA data-byte control for a 32-bit wide bus.

This transfer data size cannot exceed either of the data bus bit widths supported by the source and destination for DMA transfer. The data bus widths are fixed by the hardware.

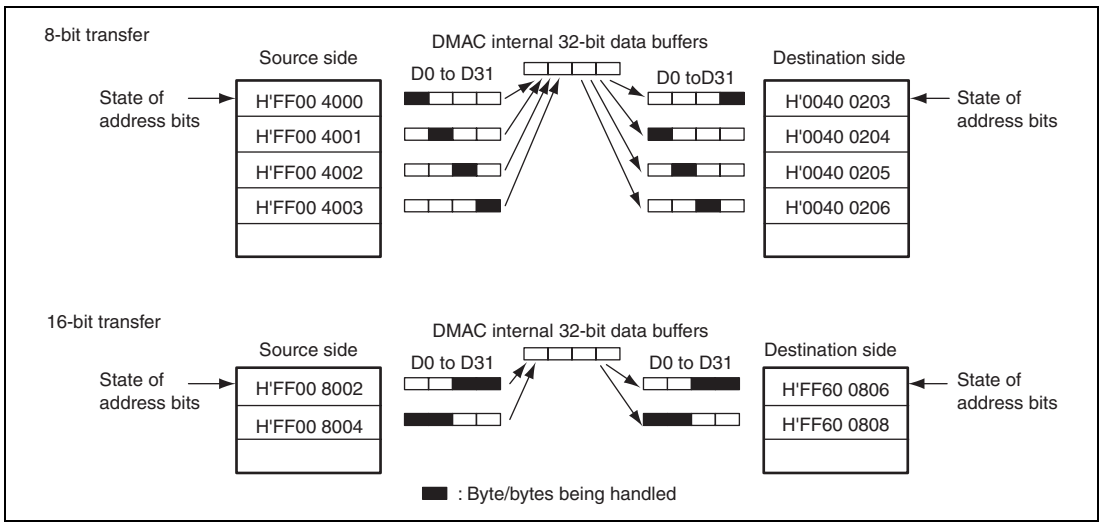


Figure 11.12 Example of DMA Data-Byte Control for 32-bit Bus Width

11.10 Reload Function

Reloading can be set up for each transfer parameter (source address, destination address, or byte count) of a channel through the setting of the individual reload function enable bits in the corresponding DMA control register A (DMCNTAn). When the DMA transfer end condition is detected, DMA transfer parameters specified for reloading are automatically reloaded.

(1) Reload and Current Registers

If reloading is not in use, only place the data in the current register. When reloading is in use, place data in both the reload and current registers.

Do not write to the current register during single operand transfer. If data is written to the register during continuous operation, further operation is not guaranteed. Although the reload register can be set during single operand transfer, ensure that this is not the last single operand transfer of a DMA transfer. If the setting is executed after that point, the new setting may not be reloaded on completion of the DMA transfer.

(2) Continuous Transfer to Dispersed areas

The reload function enables continuous transfer to dispersed areas.

Writing to the DMA reload source/destination address register (DMRSADRn/ DMRDADRn) or the DMA reload byte count register (DMRBCTn) before the completion of transfer provides a way of preparing the parameters for the next transfer without affecting the current DMA transfer (current registers). This enables the use of a single channel for the continuous transfer of multiple transfer blocks consisting of different numbers of bytes to and from different transfer areas over a single channel.

Figure 11.13 shows an example of the transfer of blocks between dispersed areas with the aid of the reload function.

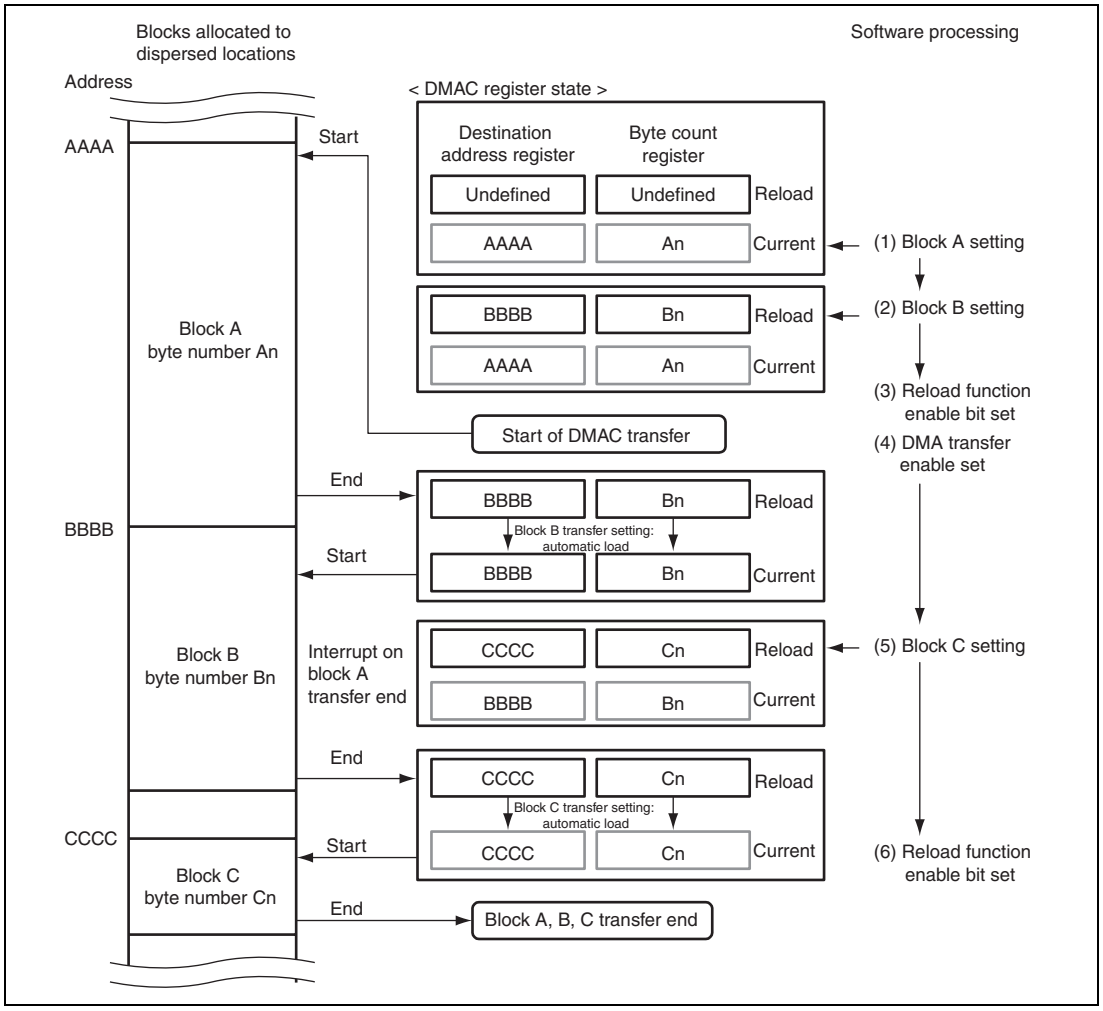


Figure 11.13 Example of Transferring Blocks between Dispersed Areas by Using the Reload Function.

11.11 Rotate Function

When rotation is selected as the address "indexing" mode, the address is incremented. On completion of single operand transfer, the value in a working source or working destination address register for which rotation has been selected returns to the value of the source or destination address register (DMCSADRn or DMCDADRn) for the corresponding channel.

Figure 11.14 is an example of transfer using the rotate function (source: rotation, destination: incrementation).

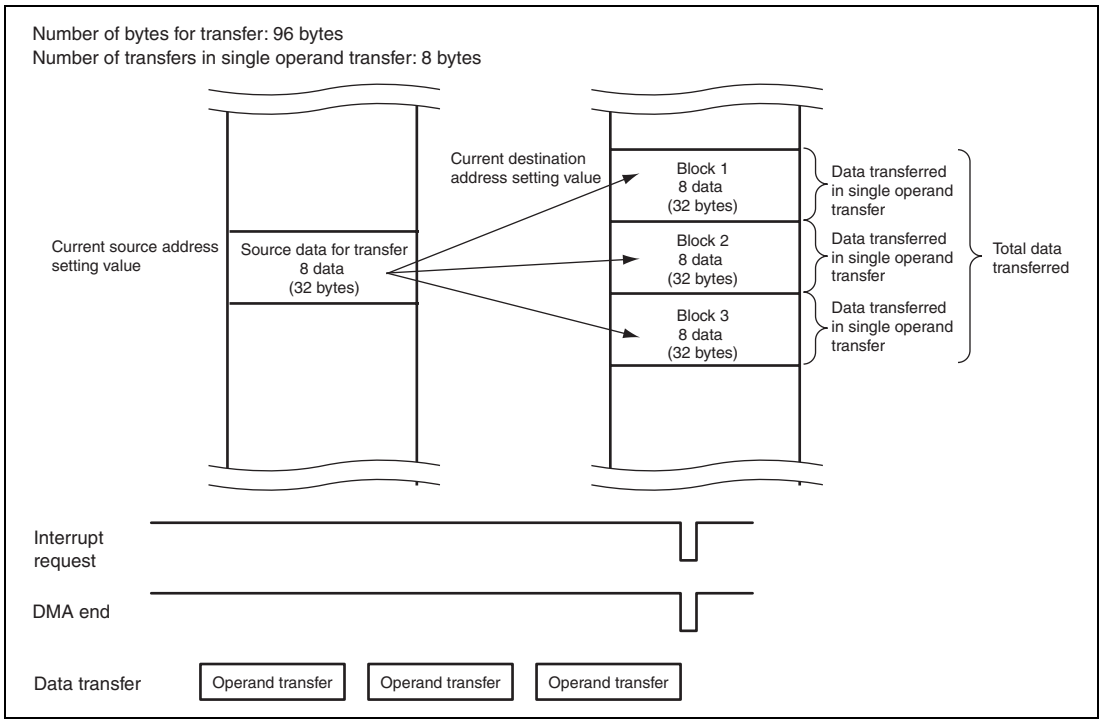


Figure 11.14 Example of Transfer Using the Rotate Function (Source: Rotate, Destination: Increment)

11.12 Transfer Speed

Transfer speeds are calculated as shown below.

(1) Conditions for Calculation

- DMA transfer mode: cycle-stealing transfer mode/pipelined transfer mode
- Transfer unit (one data size): properly aligned 32-bit data
- Operating clock: 60 MHz
- Number of cycles for access to external devices:
  four cycles for reading; and
  two cycles for writing.

(2) Formulae Used in Calculation

- Cycle-stealing transfer mode

  (data size in unit data transfer) / (number of read cycles + number of write cycles + one idle cycle) × operating clock

- Pipelined transfer mode

  (data size in unit data transfer) / (whichever is larger of number of read or write cycles) × operating clock

Note: During transfer in the pipelined transfer mode, most read and write cycles overlap.

An example of the calculation of transfer speed is given below.

(a) Transfer between On-chip RAM

Maximum speed of transfer between on-chip RAM (0 wait) and on-chip RAM (0 wait).

- Cycle-stealing transfer mode

  4 bytes / (1 read cycle + 1 write cycle + 1 idle cycle) × 60 MHz = 79.8 Mbytes/sec

- Pipelined transfer mode
  Pipelined transfer through a single BIU is not possible. See section 11.4.1 (2), Pipelined Transfer Mode.

(b) Transfer to External Devices

Maximum transfer speed from an on-chip CPU block as the source (0 wait) to an external device (2 write cycles).

- Cycle-stealing transfer mode

  4 bytes / (1 read cycle + 2 write cycles + 1 idle cycle) × 60 MHz = 60 Mbytes/sec

- Pipelined transfer mode

  4 bytes / (2 write cycles)× 60 MHz = 120 Mbytes/sec

Maximum transfer speed from an external device (4 read cycles) to an on-chip CPU block source (0 wait)

- Cycle-stealing transfer mode

  4 bytes / (4 read cycles + 1 write cycle + 1 idle cycle) × 60 MHz = 39.6 Mbytes/sec

- Pipelined transfer mode

  4 bytes / (4 read cycles)× 60 MHz = 60 Mbytes/sec

Maximum transfer speed from an external device (4 read cycles) to an external device (2 write cycles)

- Cycle-stealing transfer mode

  4 bytes / (4 read cycles + 2 write cycles + 1 idle cycle) × 60 MHz = 34.2 Mbytes/sec

- Pipelined transfer mode

  No pipelined transfer is possible between the external devices.

Note: Access to external devices is controlled by the settings of the BSC control registers. For details, see section 9, Bus State Controller (BSC).

11.13 Usage Note

11.13.1 Note on Making a Transition To Software Standby Mode or Deep Standby Mode

If the SLEEP instruction is executed to make a transition to software standby mode or deep standby mode during transfer by the DMAC, the DMAC stops its operation without waiting for the completion of the transfer. Thus, the DMA transfer is not guaranteed. Therefore, when making a transition to software standby mode or deep standby mode, wait for the completion of the DMA transfer or stop the DMA transfer to execute the SLEEP instruction.

Section 12 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.

12.1 Features

- Up to 16 pulse input/output lines and three pulse input lines
- Selection of eight counter input clocks for each channel (four clocks for channel 5)
- The following operations can be set for channels 0 to 4:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture is possible
  - Register simultaneous input/output is possible by synchronous counter operation
  - A maximum 12-phase PWM output is possible in combination with synchronous operation
- Buffer operation settable for channels 0, 3, and 4
- Phase counting mode settable independently for each of channels 1 and 2
- Cascade connection operation
- Fast access via internal 16-bit bus
- 28 interrupt sources
- Automatic transfer of register data
- A/D converter start trigger can be generated
- Module standby mode can be settable
- A total of six-phase waveform output, which includes complementary PWM output, and positive and negative phases of reset PWM output by interlocking operation of channels 3 and 4, is possible.
- AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset PWM output is settable by interlocking operation of channels 0, 3, and 4, and the selection of two types of waveform outputs (chopping and level) is possible.
- Dead time compensation counter available in channel 5
- In complementary PWM mode, interrupts at the crest and trough of the counter value and A/D converter start triggers can be skipped.

Table 12.1 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 and TCNT overflow or underflow	—
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 TADCOR A_4 and TCNT_4 • A/D converter start request at a match between TADCOR B_4 and TCNT_4	—
Interrupt skipping function	—	—	—	• Skips TGRA_3 compare match interrupts	• Skips TCIV_4 interrupts	—

[Legend]
√: Possible
—: Not possible

Figure 12.1 shows a block diagram of the MTU2.

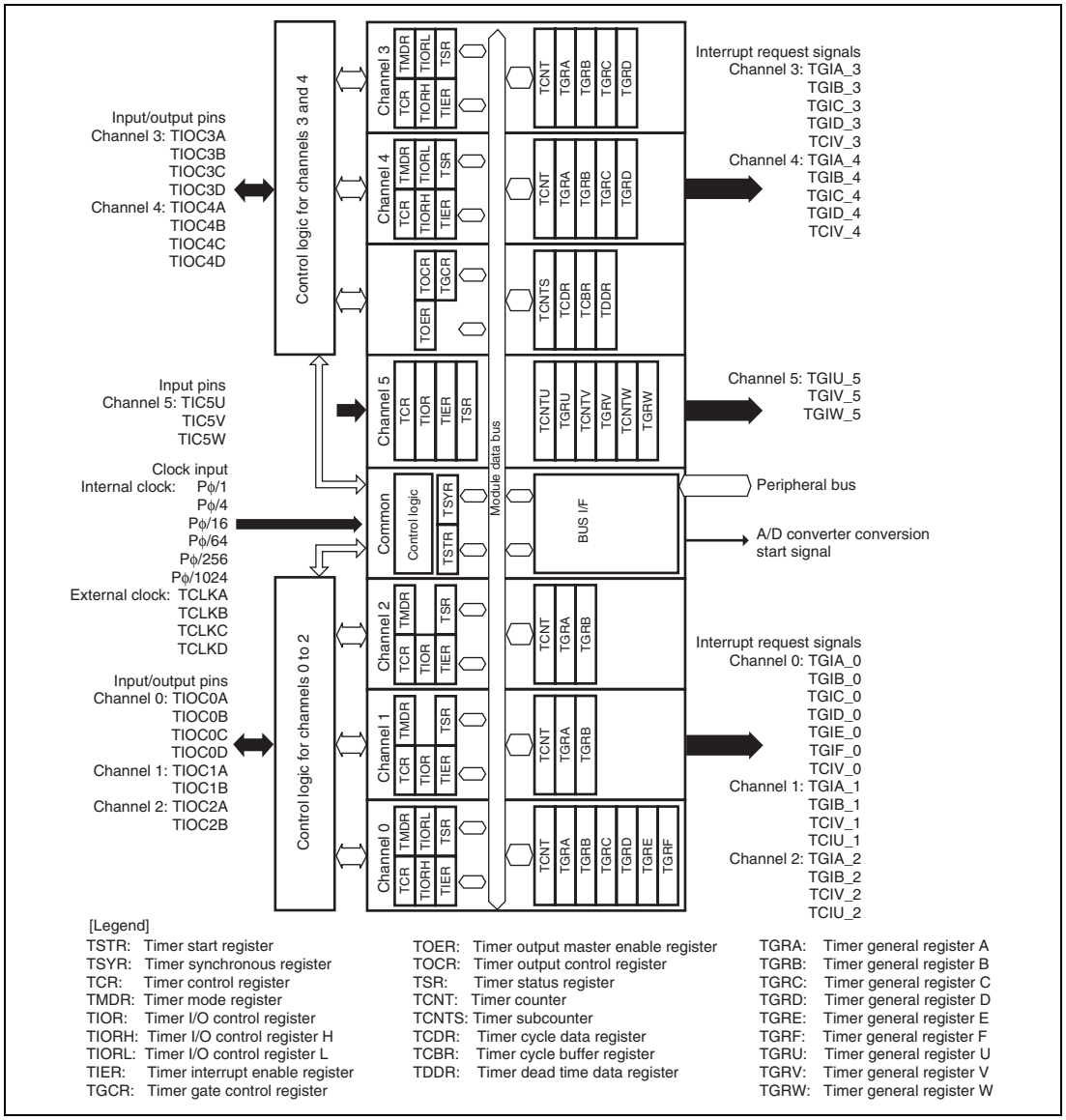


Figure 12.1 Block Diagram of MTU2

12.2 Input/Output Pins

Table 12.2 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 12.54.

12.3 Register Descriptions

The MTU2 has the following registers. For details on register addresses and register states during each process, refer to section 28, List of Registers. To distinguish registers in each channel, an underscore and the channel number are added as a suffix to the register name; TCR for channel 0 is expressed as TCR_0.

Table 12.3 Register Configuration

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, 16, 32
	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, 16
	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, 16, 32
	Timer status register_0	TSR_0	R/W	H'C0	H'FFFE4305	8
	Timer counter_0	TCNT_0	R/W	H'0000	H'FFFE4306	16
	Timer general register A_0	TGRA_0	R/W	H'FFFF	H'FFFE4308	16, 32
	Timer general register B_0	TGRB_0	R/W	H'FFFF	H'FFFE430A	16
	Timer general register C_0	TGRC_0	R/W	H'FFFF	H'FFFE430C	16, 32
	Timer general register D_0	TGRD_0	R/W	H'FFFF	H'FFFE430E	16
	Timer general register E_0	TGRE_0	R/W	H'FFFF	H'FFFE4320	16, 32
	Timer general register F_0	TGRF_0	R/W	H'FFFF	H'FFFE4322	16
	Timer interrupt enable register 2_0	TIER2_0	R/W	H'00	H'FFFE4324	8, 16
	Timer status register 2_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

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
1	Timer control register_1	TCR_1	R/W	H'00	H'FFFE4380	8, 16
	Timer mode register_1	TMDR_1	R/W	H'00	H'FFFE4381	8
	Timer I/O control register_1	TIOR_1	R/W	H'00	H'FFFE4382	8
	Timer interrupt enable register_1	TIER_1	R/W	H'00	H'FFFE4384	8, 16, 32
	Timer status register_1	TSR_1	R/W	H'C0	H'FFFE4385	8
	Timer counter_1	TCNT_1	R/W	H'0000	H'FFFE4386	16
	Timer general register A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16, 32
	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, 16
	Timer mode register_2	TMDR_2	R/W	H'00	H'FFFE4001	8
	Timer I/O control register_2	TIOR_2	R/W	H'00	H'FFFE4002	8
	Timer interrupt enable register_2	TIER_2	R/W	H'00	H'FFFE4004	8, 16, 32
	Timer status register_2	TSR_2	R/W	H'C0	H'FFFE4005	8
	Timer counter_2	TCNT_2	R/W	H'0000	H'FFFE4006	16
	Timer general register A_2	TGRA_2	R/W	H'FFFF	H'FFFE4008	16, 32
	Timer general register B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
3	Timer control register_3	TCR_3	R/W	H'00	H'FFFE4200	8, 16, 32
	Timer mode register_3	TMDR_3	R/W	H'00	H'FFFE4202	8, 16
	Timer I/O control register H_3	TIORH_3	R/W	H'00	H'FFFE4204	8, 16, 32
	Timer I/O control register L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
	Timer interrupt enable register_3	TIER_3	R/W	H'00	H'FFFE4208	8, 16
	Timer counter_3	TCNT_3	R/W	H'0000	H'FFFE4210	16, 32
	Timer general register A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16, 32
	Timer general register B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
	Timer general register C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16, 32
	Timer general register D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
	Timer status register_3	TSR_3	R/W	H'C0	H'FFFE422C	8, 16
	Timer buffer operation transfer mode register_3	TBTM_3	R/W	H'00	H'FFFE4238	8, 16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
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, 16
	Timer I/O control register L_4	TIORL_4	R/W	H'00	H'FFFE4207	8
	Timer interrupt enable register_4	TIER_4	R/W	H'00	H'FFFE4209	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, 32
	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, 32
	Timer general register D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	16
	Timer status register_4	TSR_4	R/W	H'C0	H'FFFE422D	8
	Timer buffer operation transfer mode register_4	TBTM_4	R/W	H'00	H'FFFE4239	8
	Timer A/D converter start request cycle set register A_4	TADCORA_4	R/W	H'FFFF	H'FFFE4244	16, 32
	Timer A/D converter start request cycle set register B_4	TADCORB_4	R/W	H'FFFF	H'FFFE4246	16
	Timer A/D converter start request cycle set buffer register A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE4248	16, 32
	Timer A/D converter start request cycle set buffer register B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
5	Timer counter U_5	TCNTU_5	R/W	H'0000	H'FFFE4080	16, 32
	Timer general register U_5	TGRU_5	R/W	H'FFFF	H'FFFE4082	16
	Timer control register U_5	TCRU_5	R/W	H'00	H'FFFE4084	8
	Timer I/O control register U_5	TIORU_5	R/W	H'00	H'FFFE4086	8
	Timer counter V_5	TCNTV_5	R/W	H'0000	H'FFFE4090	16, 32
	Timer general register V_5	TGRV_5	R/W	H'FFFF	H'FFFE4092	16
	Timer control register V_5	TCRV_5	R/W	H'00	H'FFFE4094	8
	Timer I/O control register V_5	TIORV_5	R/W	H'00	H'FFFE4096	8
	Timer counter W_5	TCNTW_5	R/W	H'0000	H'FFFE40A0	16, 32
	Timer general register W_5	TGRW_5	R/W	H'FFFF	H'FFFE40A2	16
	Timer control register W_5	TCRW_5	R/W	H'00	H'FFFE40A4	8
	Timer I/O control register W_5	TIORW_5	R/W	H'00	H'FFFE40A6	8
	Timer status register_5	TSR_5	R/W	H'00	H'FFFE40B0	8
	Timer interrupt enable register_5	TIER_5	R/W	H'00	H'FFFE40B2	8
	Timer start register_5	TSTR_5	R/W	H'00	H'FFFE40B4	8
	Timer compare match clear register	TCNTCMPCLR	R/W	H'00	H'FFFE40B6	8
Common	Timer start register	TSTR	R/W	H'00	H'FFFE4280	8, 16
	Timer synchronous register	TSYR	R/W	H'00	H'FFFE4281	8
	Timer counter synchronous start register	TCSYSTR	R/W	H'00	H'FFFE4282	8
	Timer read/write enable register	TRWER	R/W	H'01	H'FFFE4284	8

Channel	Register Name	Abbreviation	R/W	Initial value	Address	Access Size
Common to 3 and 4	Timer output master enable register	TOER	R/W	H'C0	H'FFFE420A	8
	Timer gate control register	TGCR	R/W	H'80	H'FFFE420D	8
	Timer output control register 1	TOCR1	R/W	H'00	H'FFFE420E	8, 16
	Timer output control register 2	TOCR2	R/W	H'00	H'FFFE420F	8
	Timer cycle data register	TCDR	R/W	H'FFFF	H'FFFE4214	16, 32
	Timer dead time data register	TDDR	R/W	H'FFFF	H'FFFE4216	16
	Timer subcounter	TCNTS	R	H'0000	H'FFFE4220	16, 32
	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, 16
	Timer interrupt skipping counter	TITCNT	R	H'00	H'FFFE4231	8
	Timer buffer transfer set register	TBTER	R/W	H'00	H'FFFE4232	8
	Timer dead time enable register	TDER	R/W	H'01	H'FFFE4234	8
	Timer output level buffer register	TOLBR	R/W	H'00	H'FFFE4236	8
	Timer A/D converter start request control register	TADCR	R/W	H'0000	H'FFFE4240	16
	Timer waveform control register	TWCR	R/W	H'00	H'FFFE4260	8

12.3.1 Timer Control Register (TCR)

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

Bit:	7	6	5	4	3	2	1	0
	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	CCLR[2:0]	000	R/W	Counter Clear 0 to 2 These bits select the TCNT counter clearing source. See tables 12.4 and 12.5 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 12.6 to 12.10 for details.

[Legend]
x: Don't care

Table 12.4 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 12.5 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 12.6 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 12.7 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 12.8 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 12.9 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 12.10 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.

12.3.2 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. When TGRF is used as a buffer register, TGRF compare match is generated. 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 other than complementary PWM mode. TGRD compare match is generated in complementary PWM mode. When compare match occurs during the tb period in complementary PWM mode, TGRD 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 other than complementary PWM mode. TGRC compare match is generated 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 12.11 for details.

Table 12.11 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.

12.3.3 Timer I/O Control Register (TIOR)

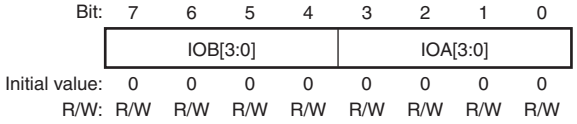
The TIOR registers are 8-bit readable/writable registers that control the TGR registers. The MTU2 has a total of eleven TIOR registers, two each for channels 0, 3, and 4, one each for channels 1 and 2, and three (TIORU_5, TIORV_5, and TIORW_5) for channel 5.

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 12.12 TIOR_1: Table 12.14 TIOR_2: Table 12.15 TIORH_3: Table 12.16 TIORH_4: Table 12.18
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 12.20 TIOR_1: Table 12.22 TIOR_2: Table 12.23 TIORH_3: Table 12.24 TIORH_4: Table 12.26

- TIORL_0, TIORL_3, TIORL_4

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	IOD[3:0]	0000	R/W	I/O Control D0 to D3 Specify the function of TGRD. See the following tables. TIORL_0: Table 12.13 TIORL_3: Table 12.17 TIORL_4: Table 12.19
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 12.21 TIORL_3: Table 12.25 TIORL_4: Table 12.27

- TIORU_5, TIORV_5, TIORW_5

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 to 0	IOC[4:0]	00000	R/W	I/O Control C0 to C4 Specify the function of TGRU_5, TGRV_5, and TGRW_5. For details, see table 12.28.

Table 12.12 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 12.13 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 12.14 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 12.15 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 12.16 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 12.17 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 12.18 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 12.19 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 12.20 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 12.21 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 12.22 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 12.23 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 12.24 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 12.25 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 12.26 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 12.27 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 12.28 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
			1	0		Measurement of low pulse width of external input signal Capture at crest
				1		Measurement of low pulse width of external input signal Capture at crest and trough
		1	0	0		Setting prohibited
				1		Measurement of high pulse width of external input signal Capture at trough
			1	0		Measurement of high pulse width of external input signal Capture at crest
				1		Measurement of high pulse width of external input signal Capture at crest and trough

[Legend]
X: Don't care

12.3.4 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

12.3.5 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 compare match between TCNTU_5 and TGRU_5. 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 compare match between TCNTV_5 and TGRV_5. 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 compare match between TCNTW_5 and TGRW_5. 0: Interrupt requests (TGIW_5) disabled 1: Interrupt requests (TGIW_5) enabled

12.3.6 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. [Setting condition] • When the TCNT value underflows (changes from H'0000 to H'FFFF) [Clearing condition] • When 0 is written to TCFU after reading TCFU = 1*2

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. [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. [Clearing condition] • When 0 is written to TCFV after reading TCFV = 1*2
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. [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 [Clearing condition] • When 0 is written to TGFD after reading TGFD = 1*2

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. [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 [Clearing condition] • When 0 is written to TGFC after reading TGFC = 1*2
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. [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 [Clearing condition] • When 0 is written to TGFB after reading TGFB = 1*2

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. [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 [Clearing conditions] • When DMAC is activated by TGIA interrupt • When 0 is written to TGFA after reading TGFA = 1*[2]

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. [Setting condition] • When TCNT_0 = TGRF_0 and TGRF_0 is functioning as compare register [Clearing condition] • When 0 is written to TGFF after reading TGFF = 1*2
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. [Setting condition] • When TCNT_0 = TGRE_0 and TGRE_0 is functioning as compare register [Clearing condition] • When 0 is written to TGFE after reading TGFE = 1*2

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. [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 while TGRU_5 is functioning as input capture register • When TCNTU_5 value is transferred to TGRU_5 while TGRU_5 is functioning as a register for measuring the pulse width of the external input signal*2. [Clearing condition] • When 0 is written to CMFU5 after reading CMFU5 = 1

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. [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 while TGRV_5 is functioning as input capture register • When TCNTV_5 value is transferred to TGRV_5 while TGRV_5 is functioning as a register for measuring the pulse width of the external input signal*2. [Clearing condition] • When 0 is written to CMFV5 after reading CMFV5 = 1
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. [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 while TGRW_5 is functioning as input capture register • When TCNTW_5 value is transferred to TGRW_5 while TGRW_5 is functioning as a register for measuring the pulse width of the external input signal*2. [Clearing condition] • When 0 is written to CMFW5 after reading CMFW5 = 1

Notes: 1. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.
2. Timing to 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).

12.3.7 Timer Buffer Operation Transfer Mode Register (TBTM)

TBTM is an 8-bit readable/writable register that specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The 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. For channels 3 and 4, bit 2 is reserved. It is always read as 0 and the write value should always be 0. Do not set this bit to 1 when channel 0 is to be used in a mode other than PWM mode. 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. Do not set this bit to 1 when the channel is to be used in a mode other than PWM mode. 0: When compare match B occurs in each channel 1: When TCNT is cleared in each channel

Bit	Bit Name	Initial Value	R/W	Description
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. Do not set this bit to 1 when the channel is to be used in a mode other than PWM mode. 0: When compare match A occurs in each channel 1: When TCNT is cleared in each channel

12.3.8 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

Bit	Bit Name	Initial Value	R/W	Description
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
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

12.3.9 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 12.29.

Bit	Bit Name	Initial Value	R/W	Description
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
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

Bit	Bit Name	Initial Value	R/W	Description
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
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 12.29 Setting of Transfer Timing by BF1 and BF0 Bits

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.

12.3.10 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.

12.3.11 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.

12.3.12 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.

12.3.13 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.

12.3.14 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 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
6	CST3	0	R/W	
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 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_2 to TCNT_0 count operation is stopped 1: TCNT_2 to TCNT_0 performs count operation
1	CST1	0	R/W	
0	CST0	0	R/W	

- 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

12.3.15 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	

12.3.16 Timer Counter Synchronous Start Register (TCSYSTR)

TCSYSTR is an 8-bit readable/writable register that specifies synchronous start of the MTU2 counters.

Bit:	7	6	5	4	3	2	1	0
	SCH0	SCH1	SCH2	SCH3	SCH4	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R

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
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

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

Note: * Only 1 can be written to set the register.

12.3.17 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 TCNT_4.

12.3.18 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 channel 3 and channel 4 prior to setting TIOR of channel 3 and channel 4.

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 12.3.19, Timer Output Control Register 1 (TOCR1), and section 12.3.20, 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. If these bits are set to 0, low level is output.

12.3.19 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)*1	R/W	R/W	R/W

Note: 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.

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 12.30.
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 12.31.

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 TOCS bit to 0 makes this bit setting valid.

Table 12.30 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 12.31 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 12.2 shows an example of complementary PWM mode output (1 phase) when OLSN = 1 and OLSP = 1.

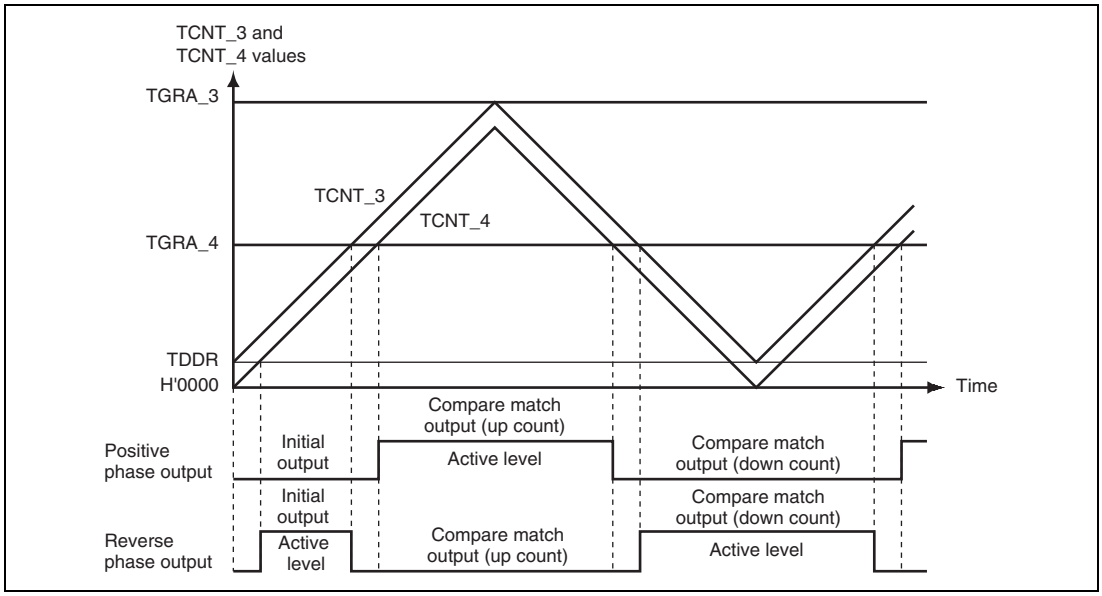


Figure 12.2 Complementary PWM Mode Output Level Example

12.3.20 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 12.32.
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 12.33.
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 12.34.
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 12.35.
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 12.36.
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 12.37.

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 12.38.

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

Table 12.32 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 12.33 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 12.34 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 12.35 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 12.36 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 12.37 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 12.38 TIOC3B Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLS1P	Initial Output	Active Level	Up Count	Down Count
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

12.3.21 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 12.3 shows an example of the PWM output level setting procedure in buffer operation.

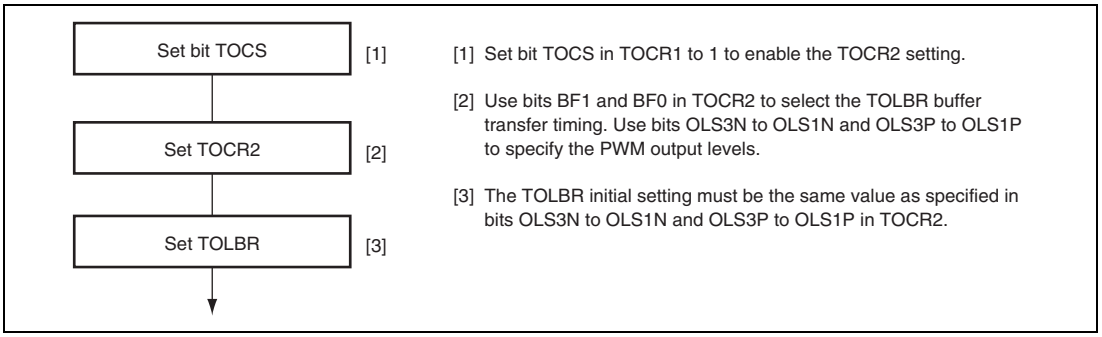


Figure 12.3 PWM Output Level Setting Procedure in Buffer Operation

12.3.22 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 (TGCR's UF, VF, WF settings).
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 12.39.
1	VF	0	R/W	
0	UF	0	R/W	

Table 12.39 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

12.3.23 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.

12.3.24 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.

12.3.25 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.

12.3.26 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.

The initial value of TCBR 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 TCBR in 8-bit units is prohibited. Always access in 16-bit units.

12.3.27 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 12.40.
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 12.41.

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 (TITCNT).

Table 12.40 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 12.41 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.

12.3.28 Timer Interrupt Skipping Counter (TITCNT)

TITCNT is an 8-bit readable/writable counter. The MTU2 has one TITCNT. TITCNT retains the value even after TCNT_3 or TCNT_4 stops counting.

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: Clear the T3AEN and T4VEN bits in TITCR to 0, to clear the value of TITCNT.

12.3.29 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 12.42.

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

Table 12.42 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

Notes: 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 12.4.8, 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.

12.3.30 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.

12.3.31 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	—	—	—	—	—	—	WRE
Initial value:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R	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 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	WRE	0	R/(W)	Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 12.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Note: * Do not set to 1 when complementary PWM mode is not selected.

12.3.32 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.

12.4 Operation

12.4.1 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 12.4 shows an example of the count operation setting procedure.

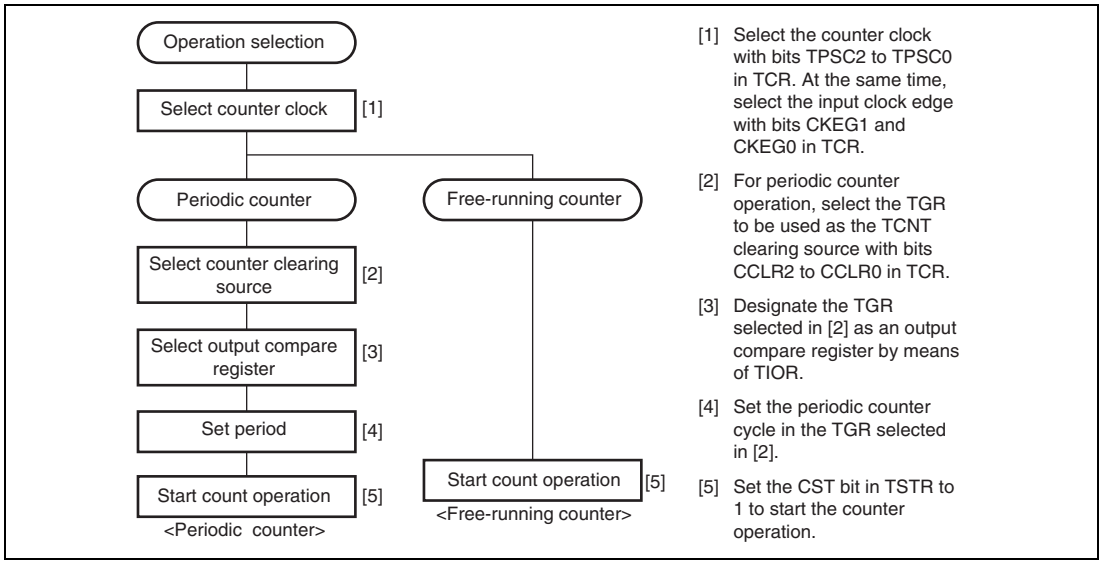


Figure 12.4 Example of Counter Operation Setting Procedure

(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 12.5 illustrates free-running counter operation.

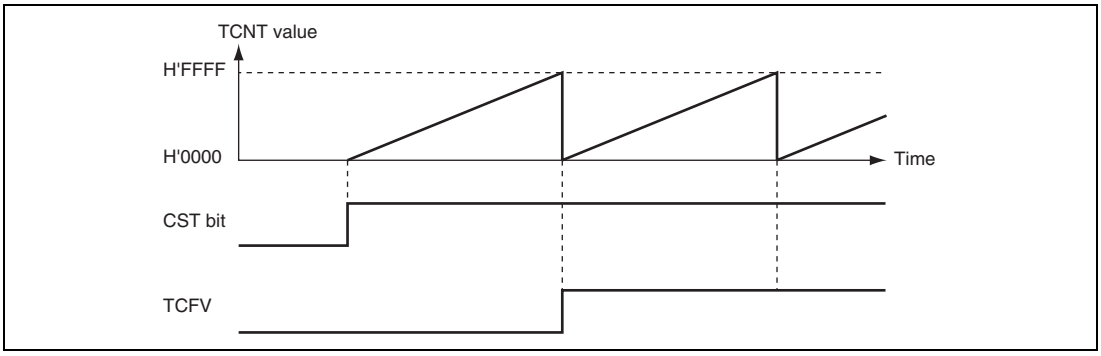


Figure 12.5 Free-Running Counter Operation

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 12.6 illustrates periodic counter operation.

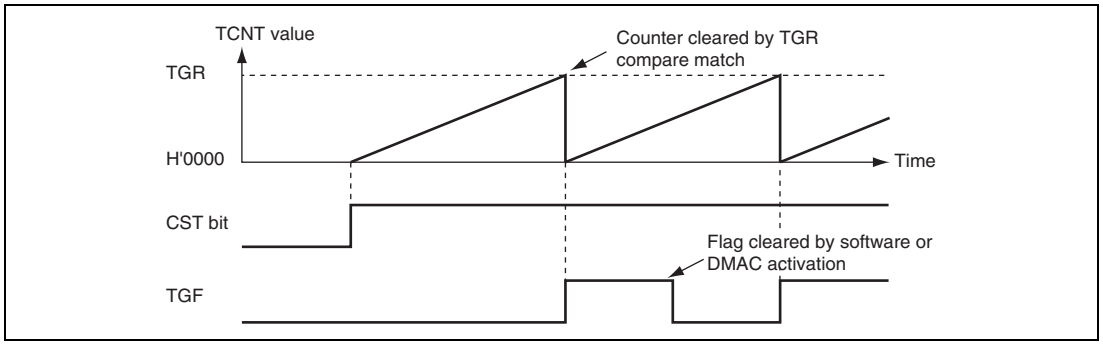


Figure 12.6 Periodic Counter Operation

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

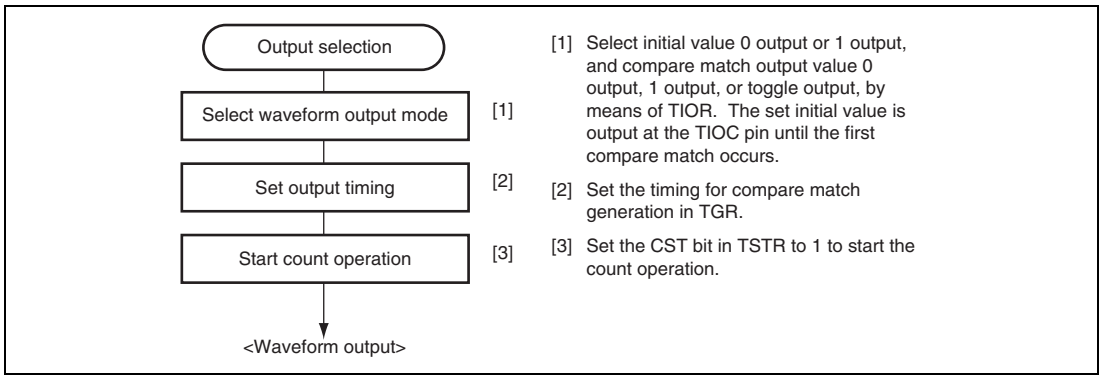


Figure 12.7 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of Waveform Output Operation:

Figure 12.8 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.

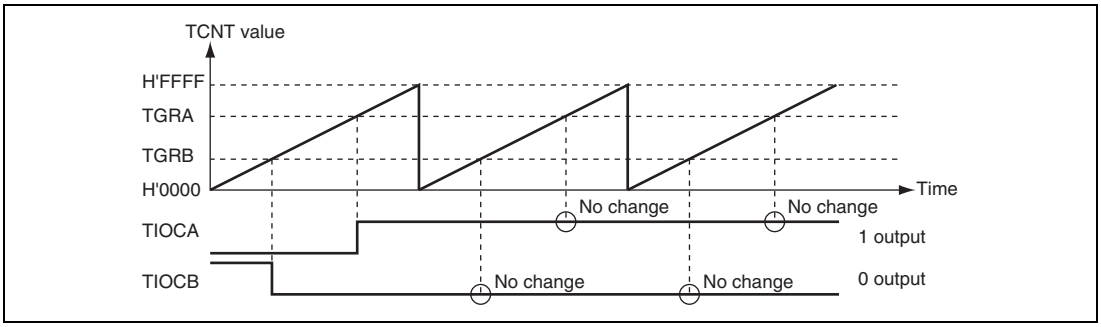


Figure 12.8 Example of 0 Output/1 Output Operation

Figure 12.9 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.

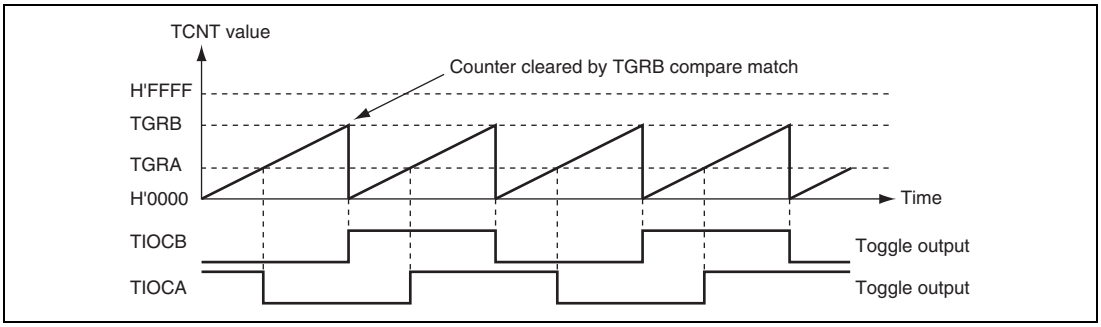


Figure 12.9 Example of Toggle Output Operation

(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 12.10 shows an example of the input capture operation setting procedure.

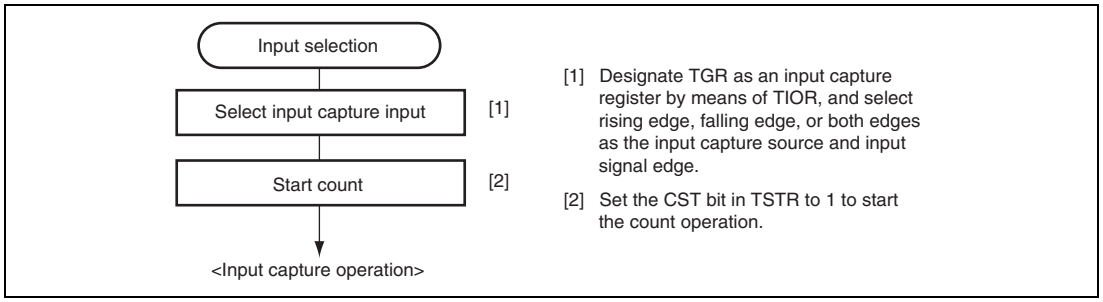


Figure 12.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation:

Figure 12.11 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.

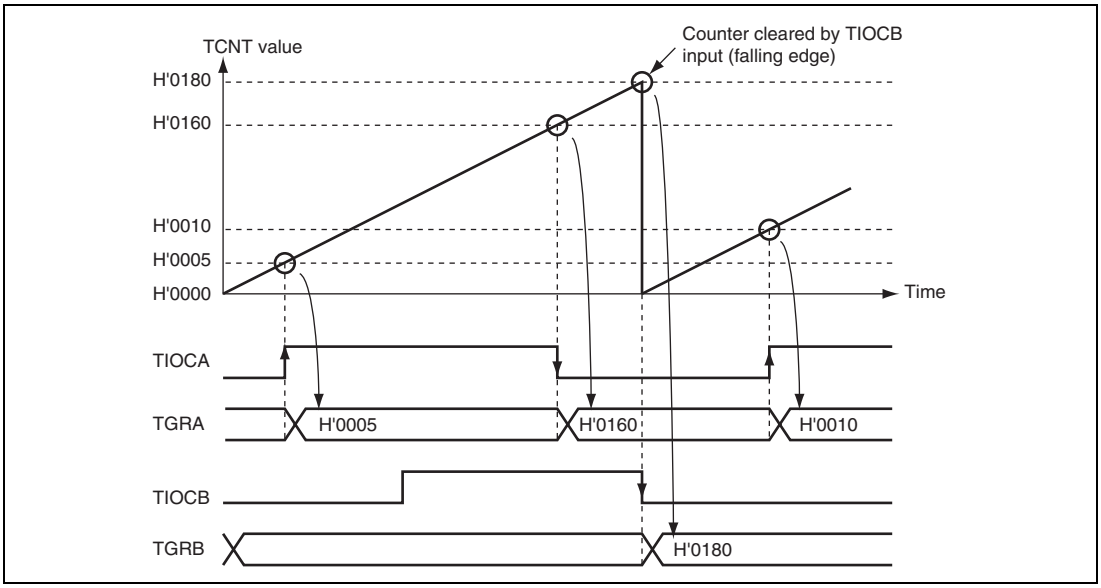


Figure 12.11 Example of Input Capture Operation

12.4.2 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 12.12 shows an example of the synchronous operation setting procedure.

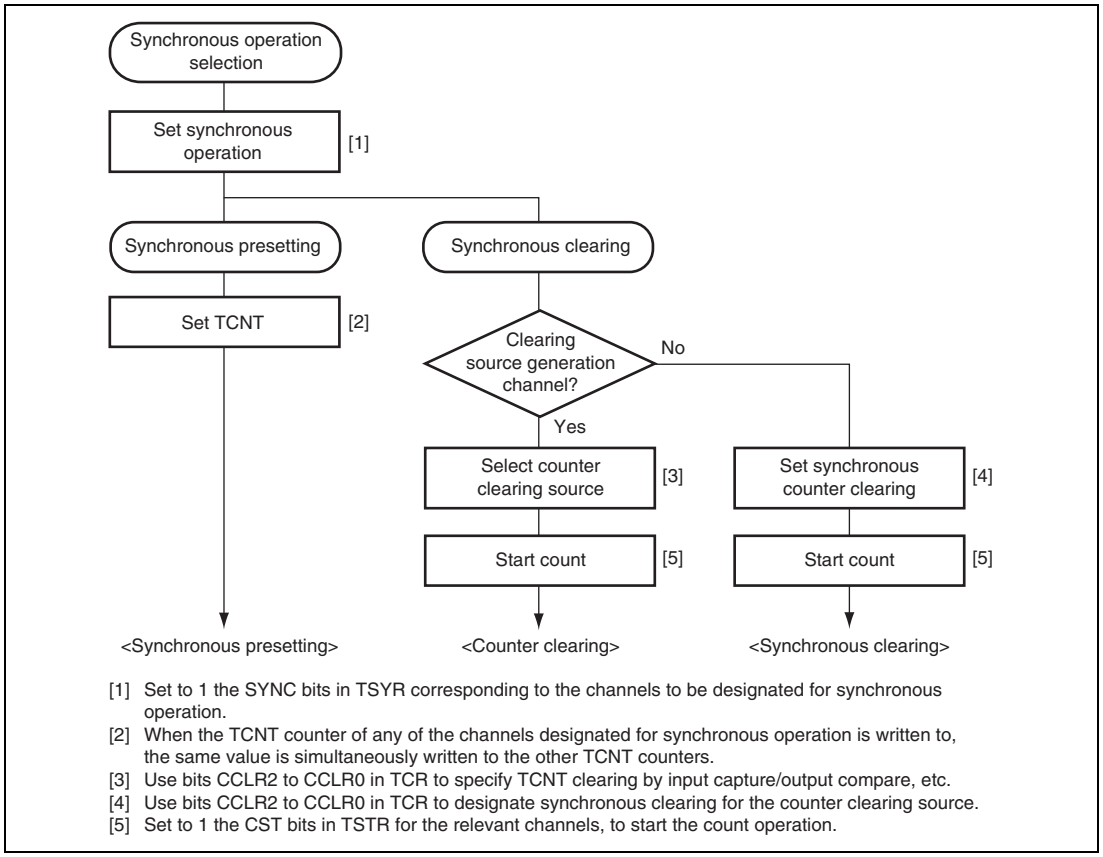


Figure 12.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 12.13 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 12.4.5, PWM Modes.

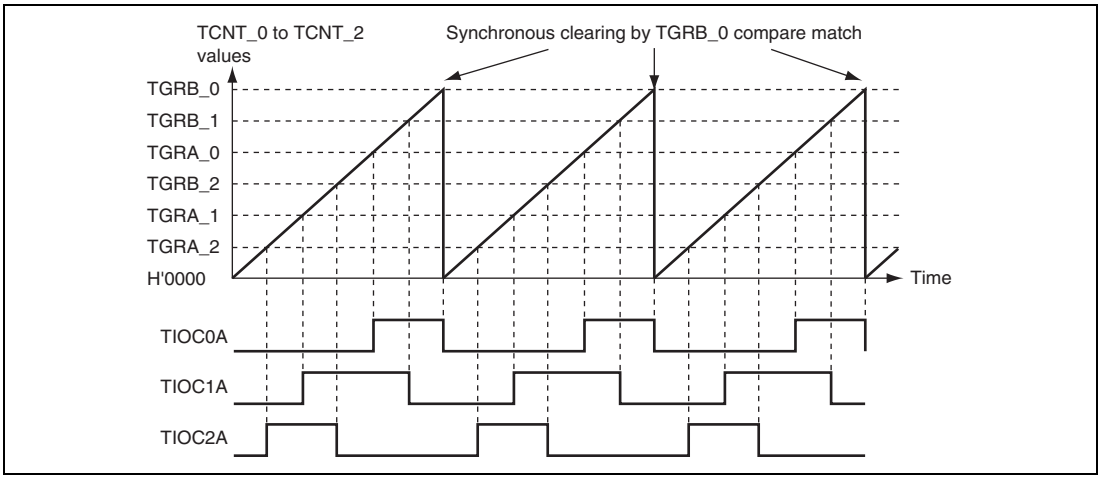


Figure 12.13 Example of Synchronous Operation

12.4.3 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 12.43 shows the register combinations used in buffer operation.

Table 12.43 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 12.14.

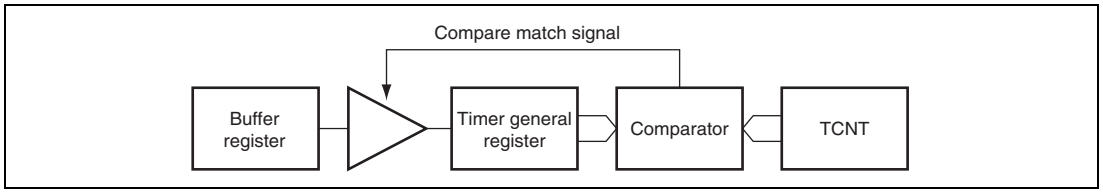


Figure 12.14 Compare Match Buffer Operation

- 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 12.15.

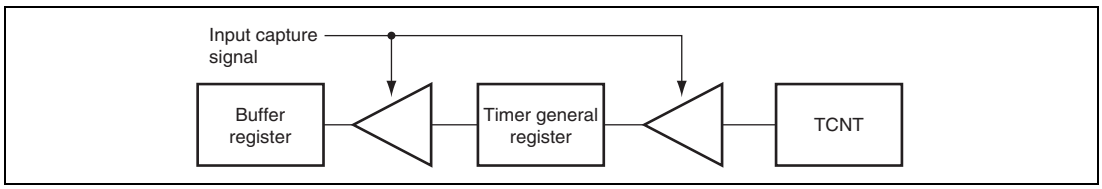


Figure 12.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 12.16 shows an example of the buffer operation setting procedure.

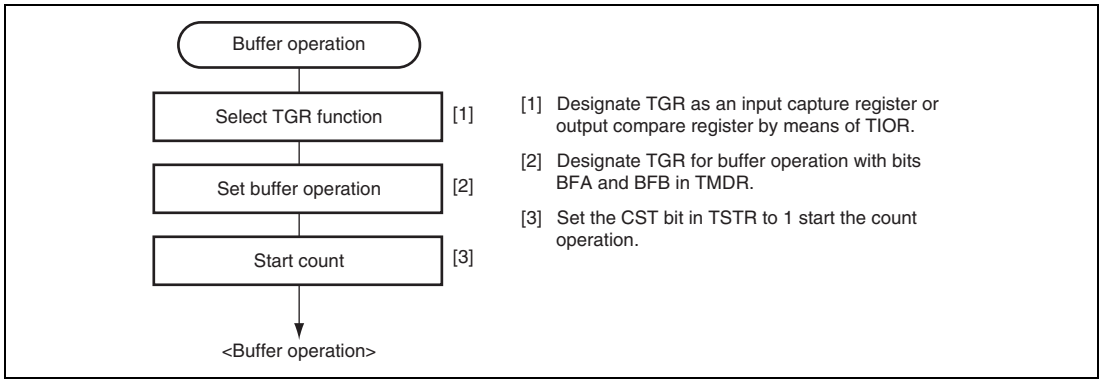


Figure 12.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an output compare register

Figure 12.17 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 12.4.5, PWM Modes.

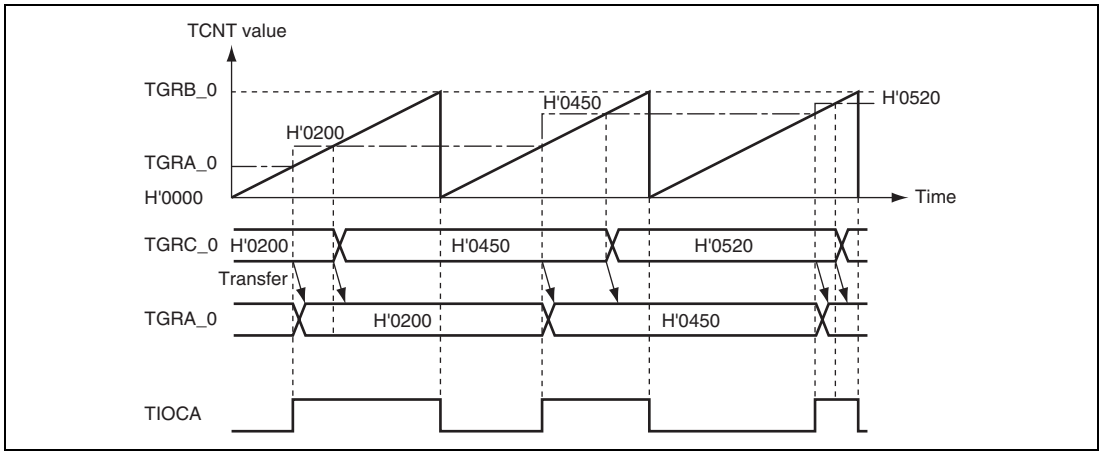


Figure 12.17 Example of Buffer Operation (1)

(b) When TGR is an input capture register

Figure 12.18 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.

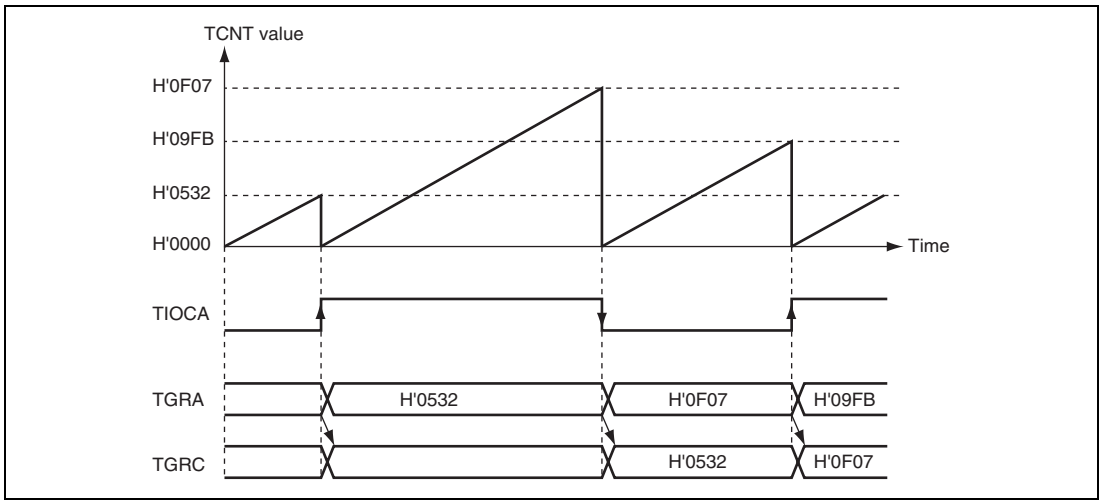


Figure 12.18 Example of Buffer Operation (2)

(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 12.19 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.

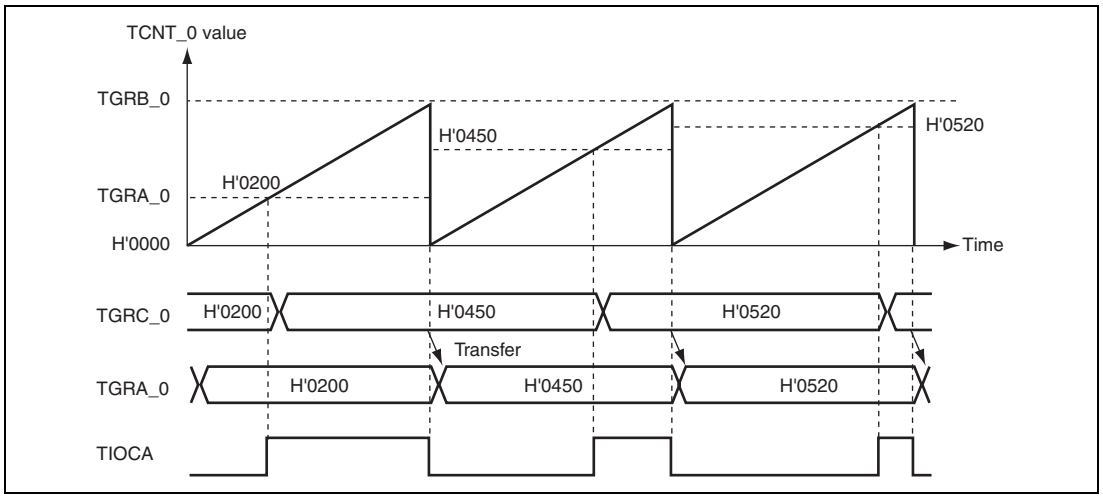


Figure 12.19 Example of Buffer Operation When TCNT_0 Clearing is Selected for TGRC_0 to TGRA_0 Transfer Timing

12.4.4 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 12.44 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 12.44 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 12.7.22, Simultaneous Capture of TCNT_1 and TCNT_2 in Cascade Connection.

Table 12.45 show the TICCR setting and input capture input pins.

Table 12.45 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 12.20 shows an example of the setting procedure for cascaded operation.

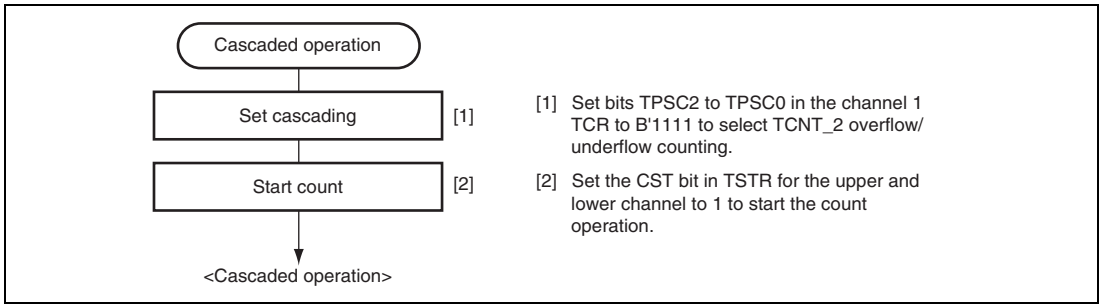


Figure 12.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 12.21 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.

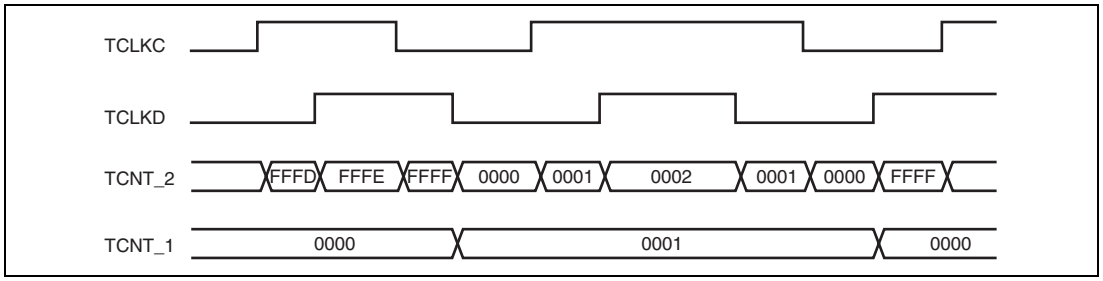


Figure 12.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 12.22 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.

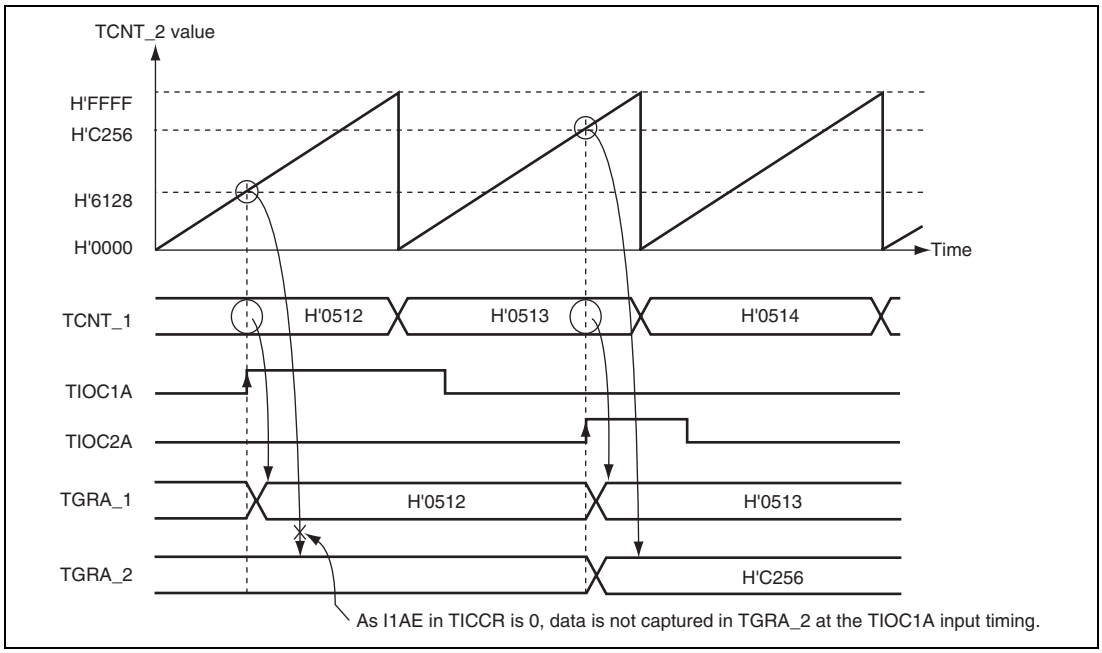


Figure 12.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 12.23 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.

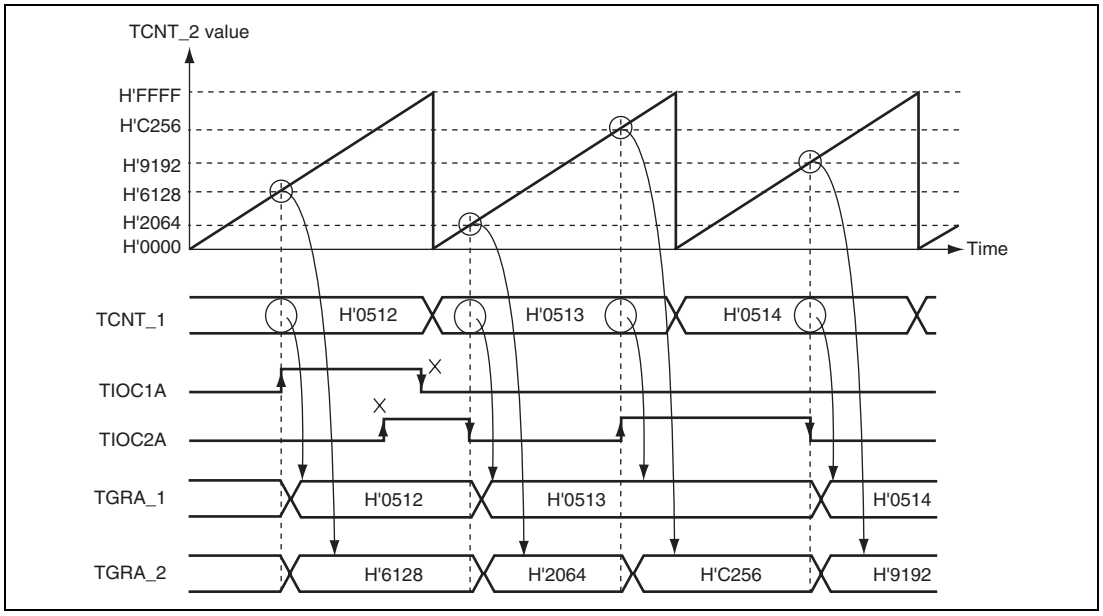


Figure 12.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 12.24 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.

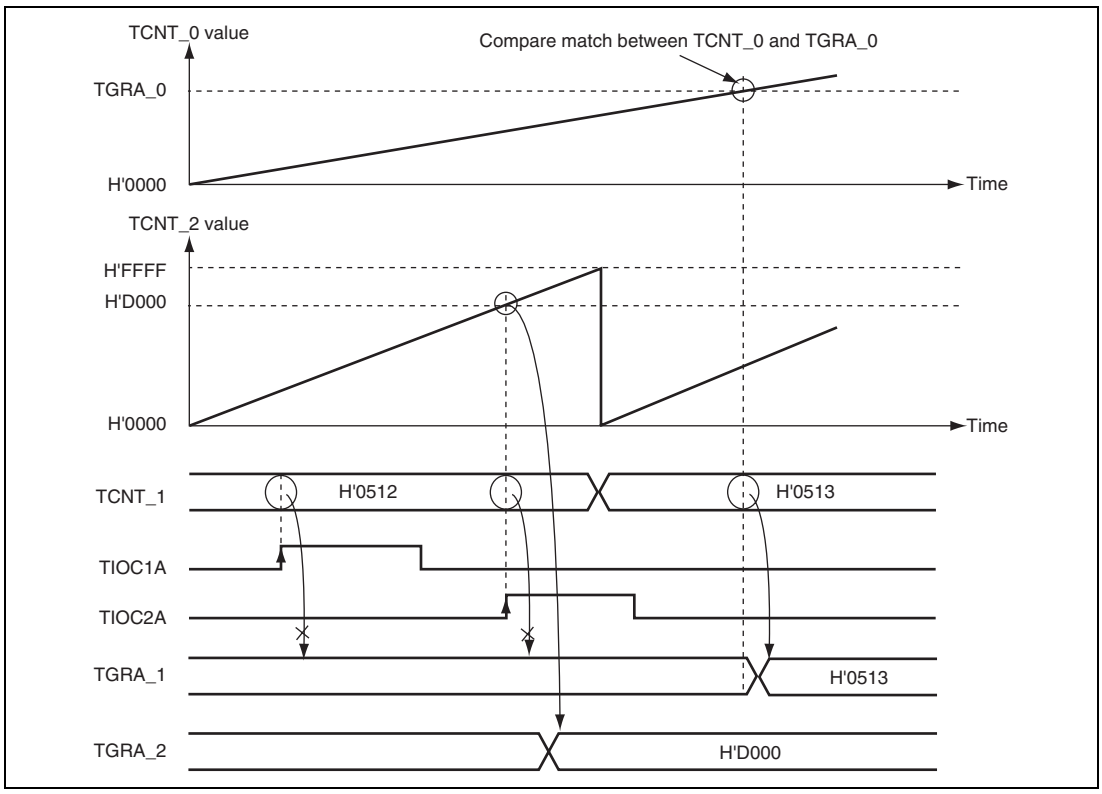


Figure 12.24 Cascaded Operation Example (d)

12.4.5 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.

1. 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.

2. 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 12.46.

Table 12.46 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 12.25 shows an example of the PWM mode setting procedure.

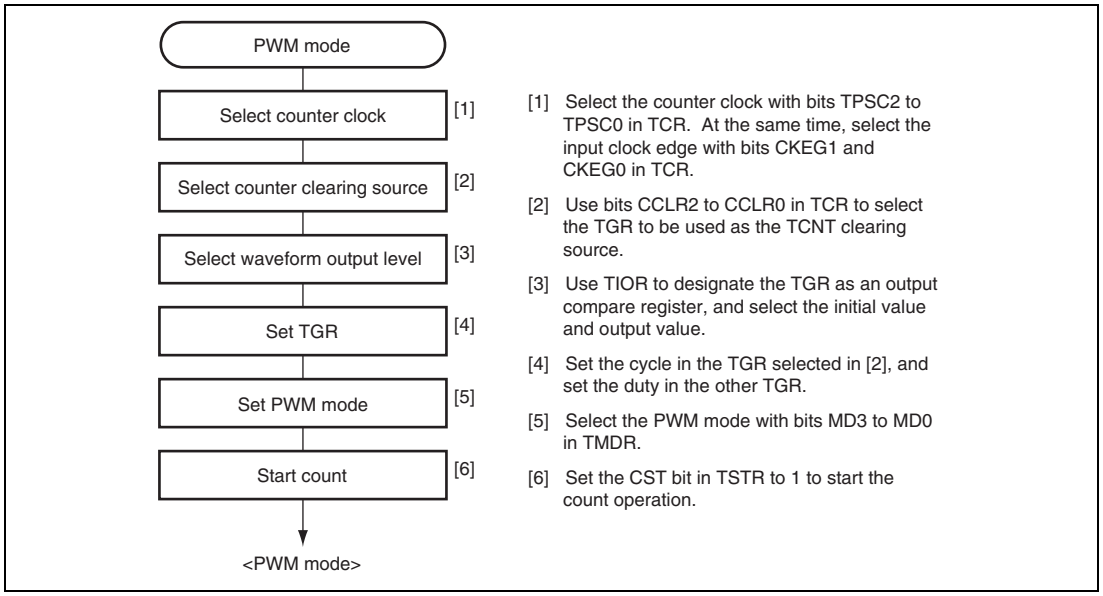


Figure 12.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 12.26 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.

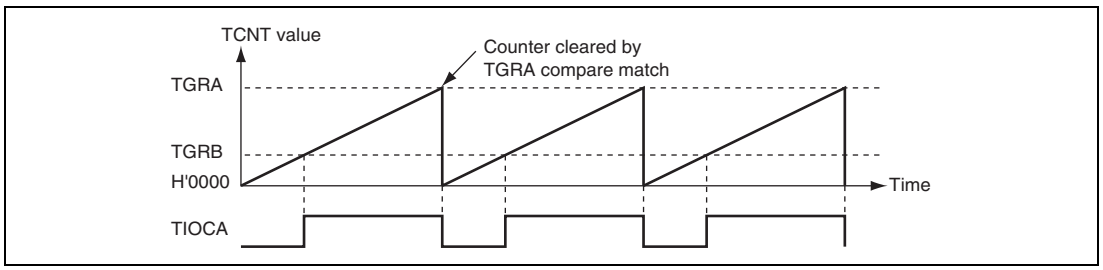


Figure 12.26 Example of PWM Mode Operation (1)

Figure 12.27 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.

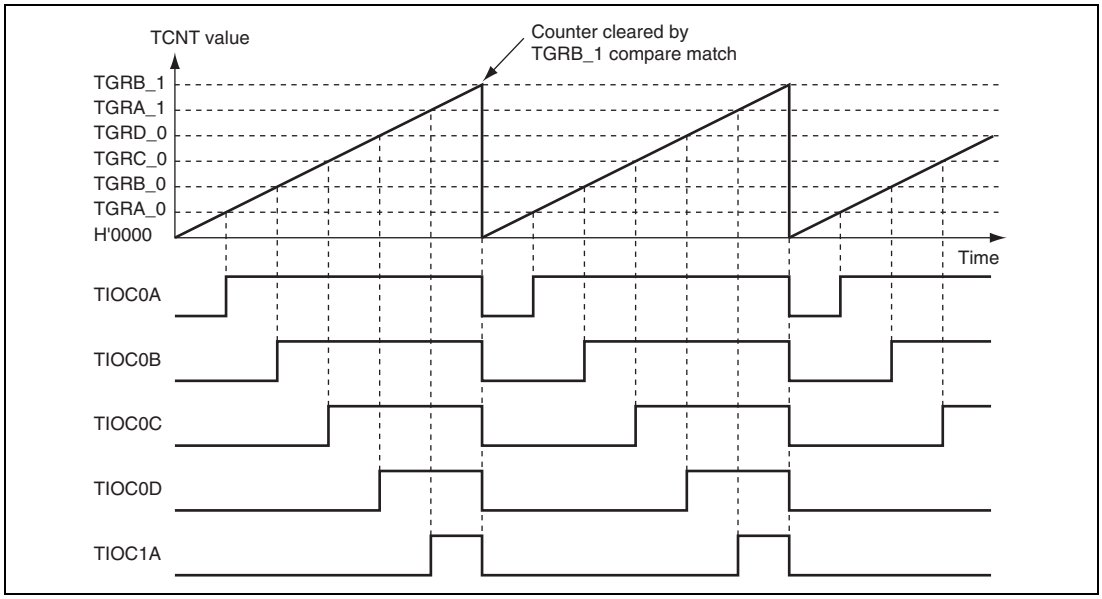


Figure 12.27 Example of PWM Mode Operation (2)

Figure 12.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

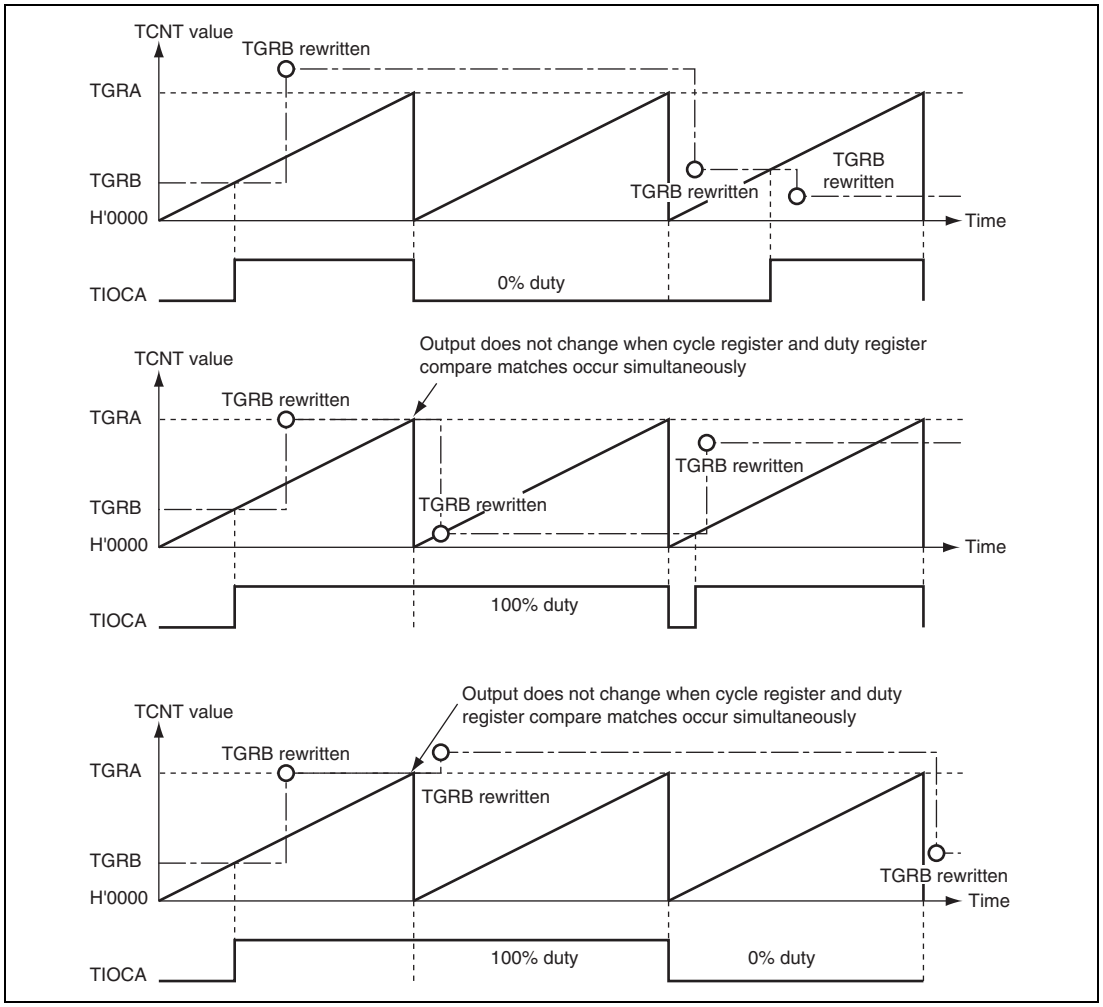


Figure 12.28 Example of PWM Mode Operation (3)

12.4.6 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 12.47 shows the correspondence between external clock pins and channels.

Table 12.47 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 12.29 shows an example of the phase counting mode setting procedure.

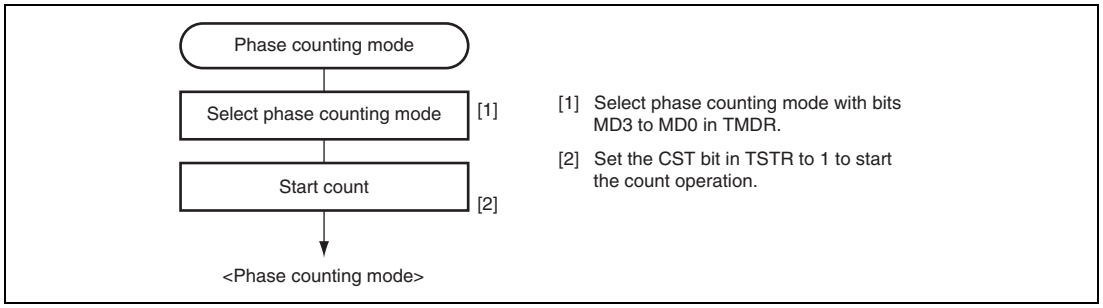


Figure 12.29 Example of Phase Counting Mode Setting Procedure

(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 12.30 shows an example of phase counting mode 1 operation, and table 12.48 summarizes the TCNT up/down-count conditions.

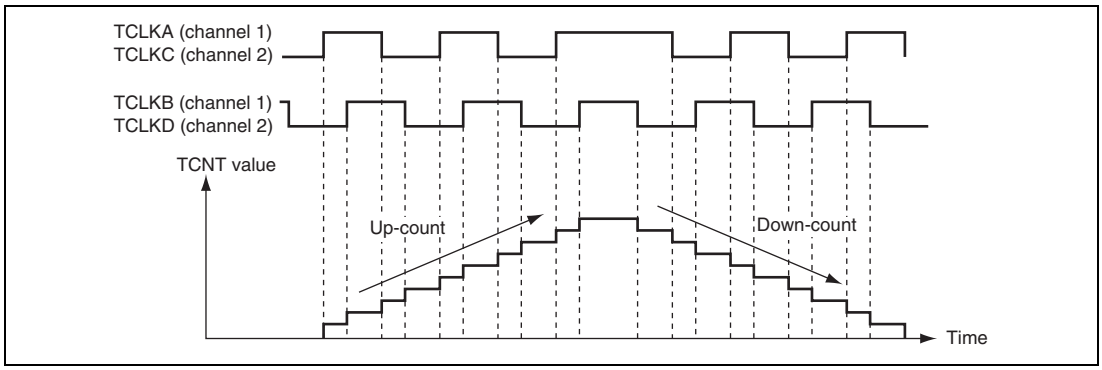


Figure 12.30 Example of Phase Counting Mode 1 Operation

Table 12.48 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 12.31 shows an example of phase counting mode 2 operation, and table 12.49 summarizes the TCNT up/down-count conditions.

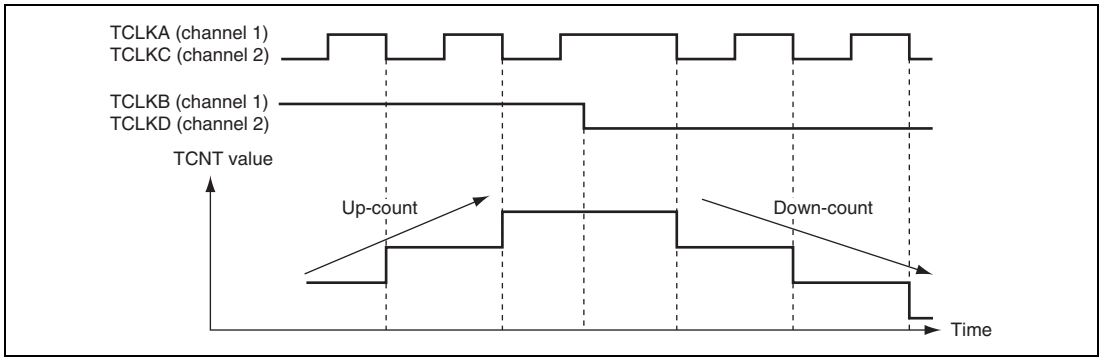


Figure 12.31 Example of Phase Counting Mode 2 Operation

Table 12.49 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 12.32 shows an example of phase counting mode 3 operation, and table 12.50 summarizes the TCNT up/down-count conditions.

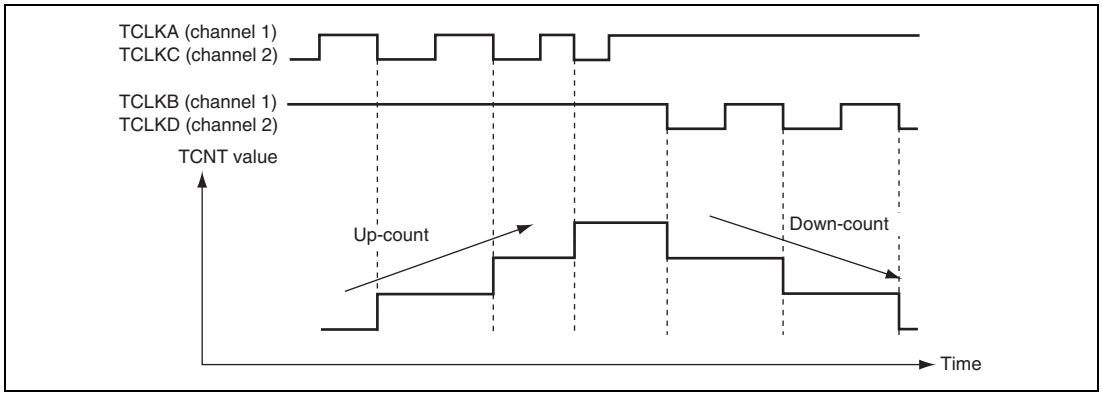


Figure 12.32 Example of Phase Counting Mode 3 Operation

Table 12.50 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 12.33 shows an example of phase counting mode 4 operation, and table 12.51 summarizes the TCNT up/down-count conditions.

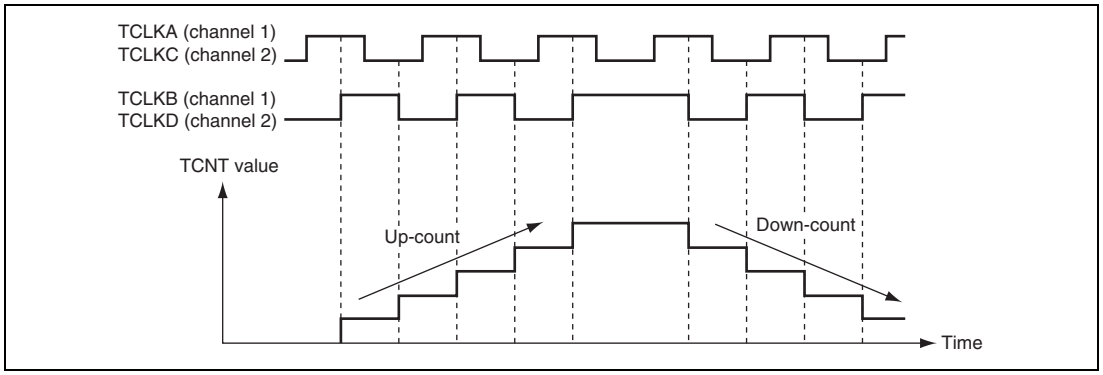


Figure 12.33 Example of Phase Counting Mode 4 Operation

Table 12.51 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 12.34 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.

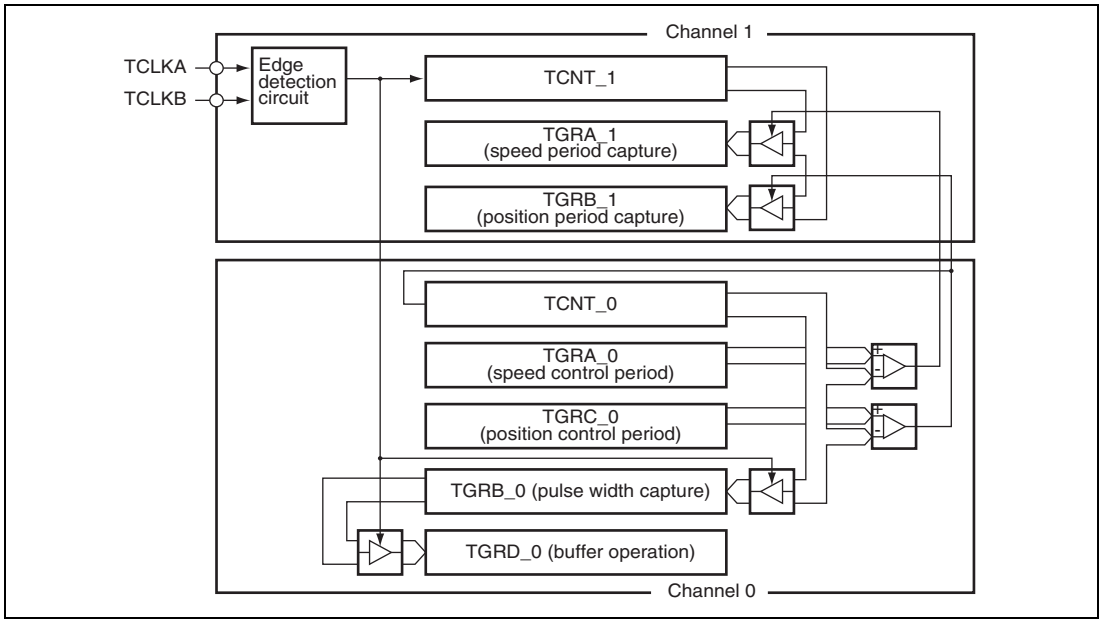


Figure 12.34 Phase Counting Mode Application Example

12.4.7 Reset-Synchronized PWM Mode

In the 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 TCNT_3 functions as an upcounter.

Table 12.52 shows the PWM output pins used. Table 12.53 shows the settings of the registers.

Table 12.52 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 12.53 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 12.35 shows an example of procedure for selecting the reset synchronized PWM mode.

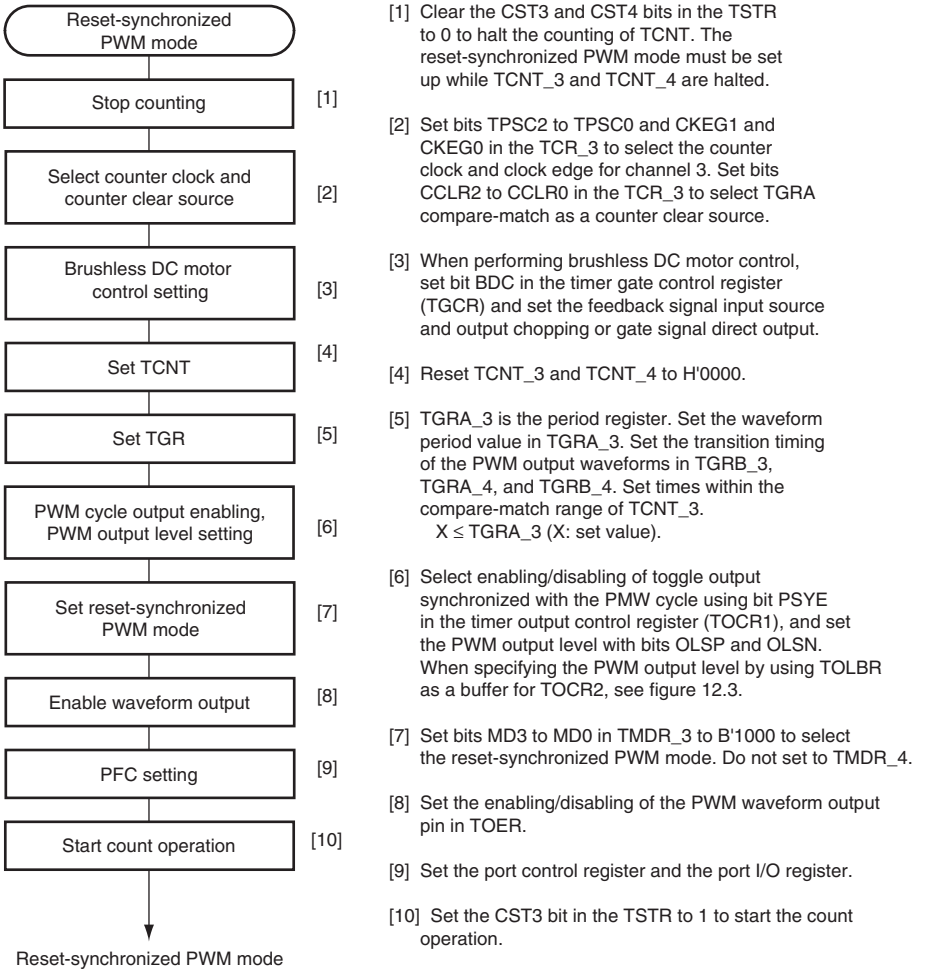


Figure 12.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Reset-Synchronized PWM Mode Operation

Figure 12.36 shows an example of operation in the 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.

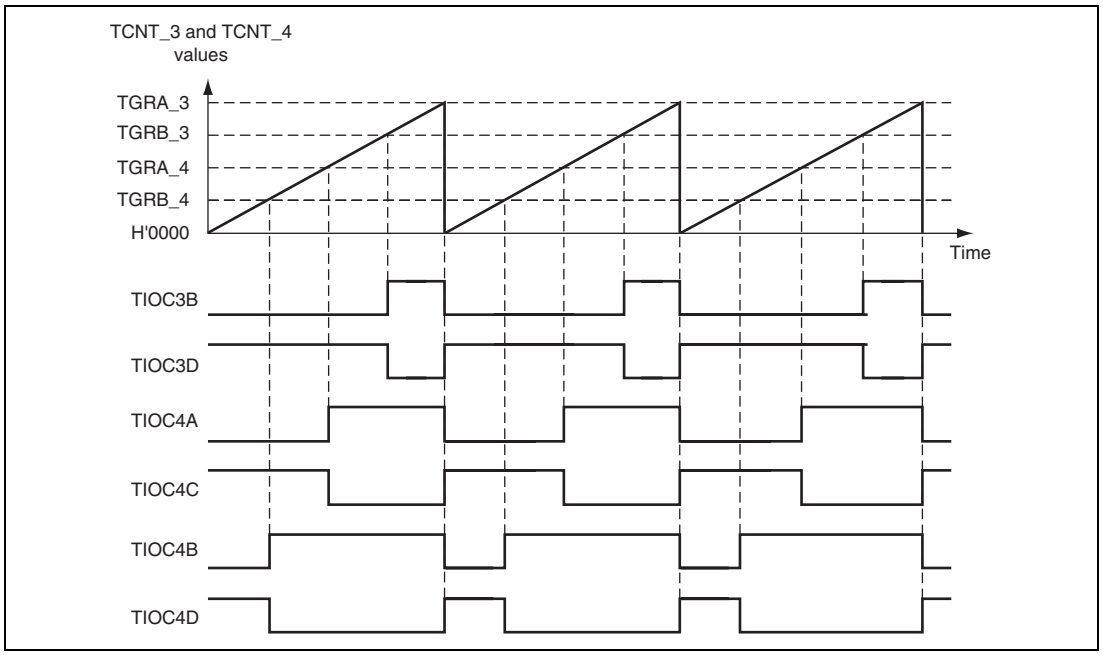


Figure 12.36 Reset-Synchronized PWM Mode Operation Example (When TOCR's OLSN = 1 and OLSP = 1)

12.4.8 Complementary PWM Mode

In the 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 is 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 12.54 shows the PWM output pins used. Table 12.55 shows the settings of the registers used.

Table 12.54 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 the complementary PWM mode.

Table 12.55 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).

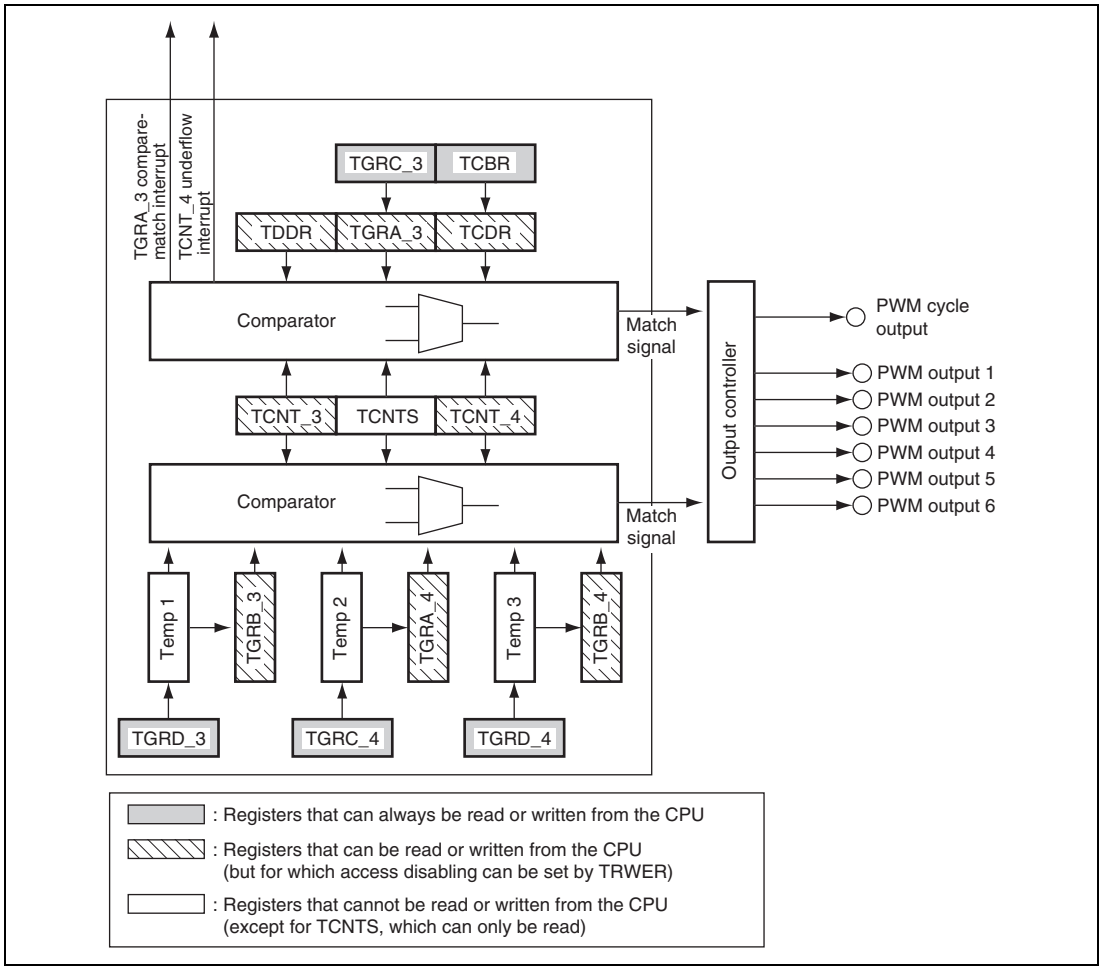


Figure 12.37 Block Diagram of Channels 3 and 4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

An example of the complementary PWM mode setting procedure is shown in figure 12.38.

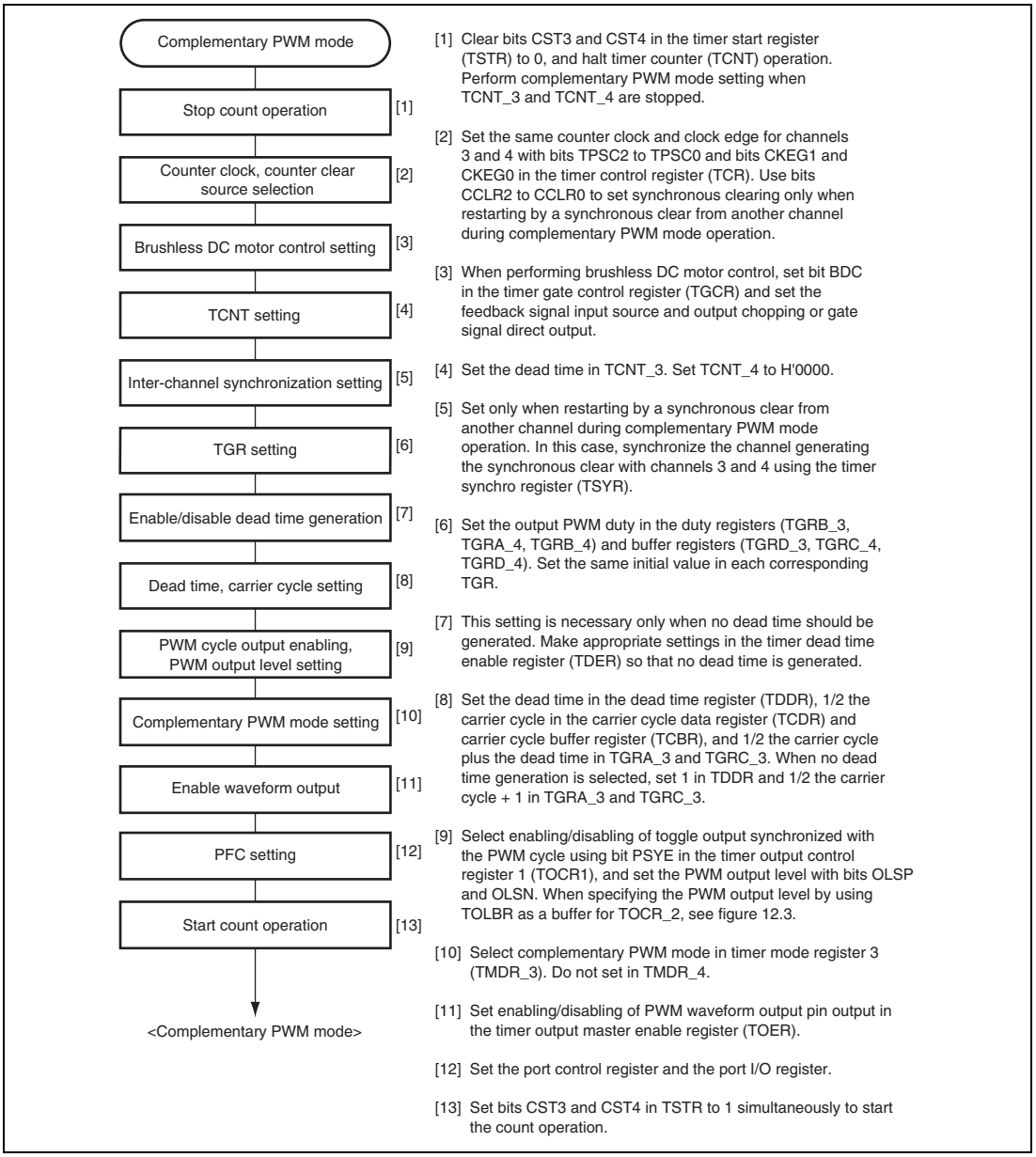


Figure 12.38 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, 6-phase PWM output is possible. Figure 12.39 illustrates counter operation in complementary PWM mode, and figure 12.40 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, TCNT_4 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.

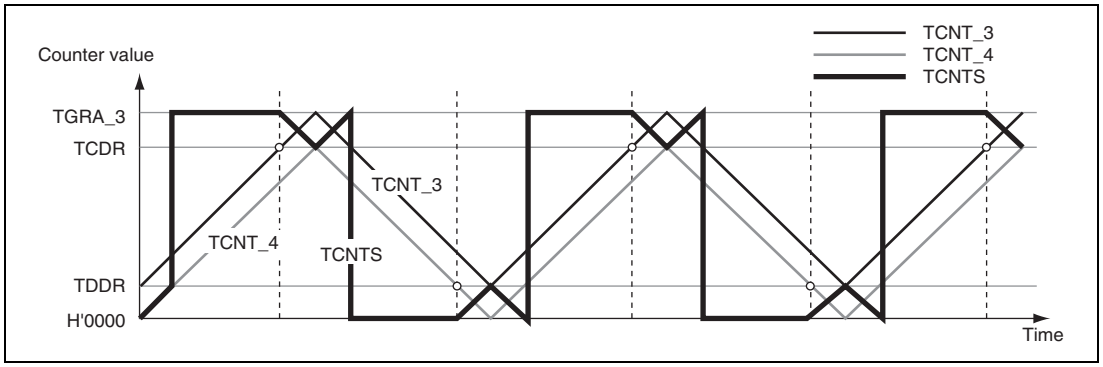


Figure 12.39 Complementary PWM Mode Counter Operation

(b) Register Operation

In complementary PWM mode, nine registers are used, comprising compare registers, buffer registers, and temporary registers. Figure 12.40 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 12.40 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 12.40) 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.

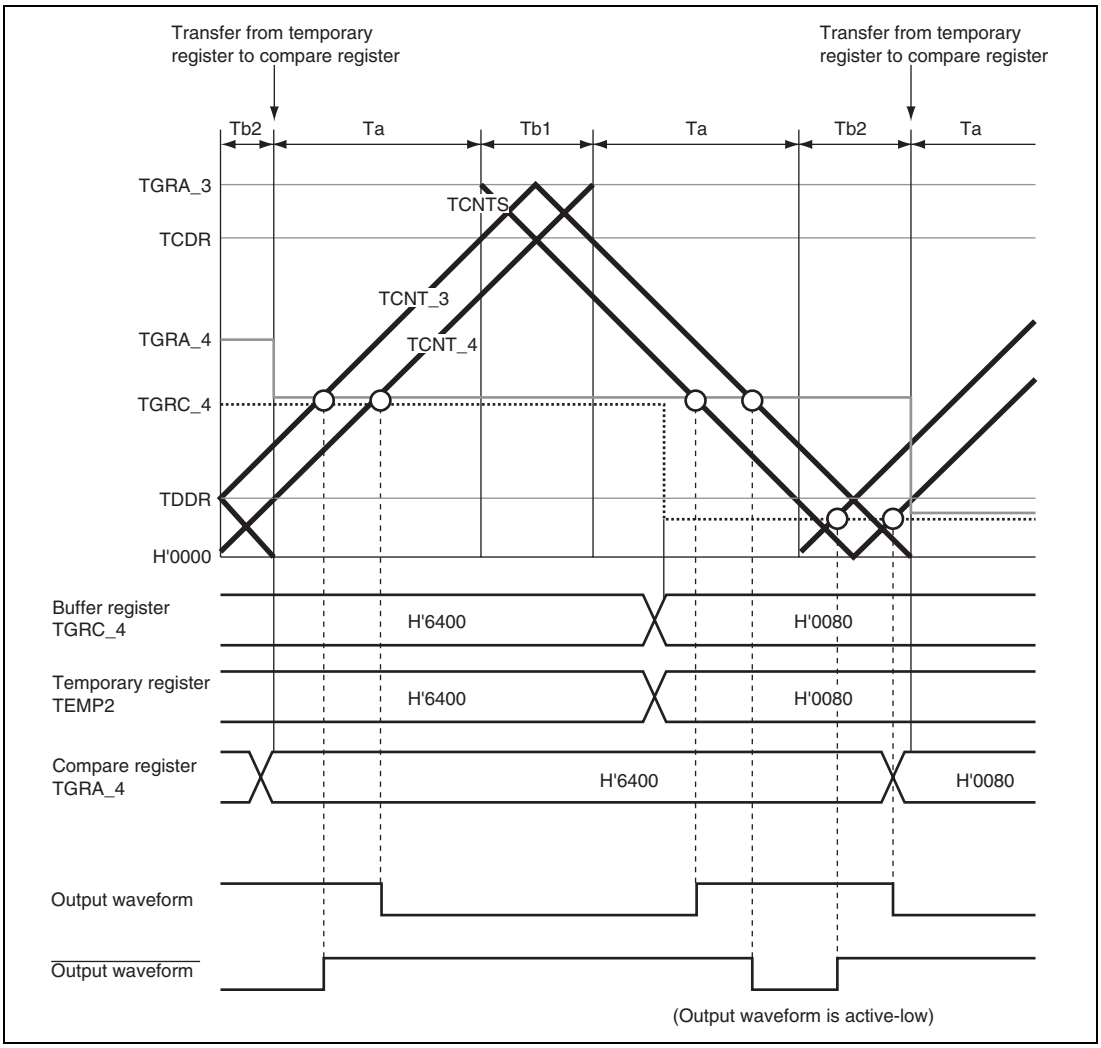


Figure 12.40 Example of Complementary PWM Mode Operation

(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 12.56 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 12.41 shows an example of operation without dead time.

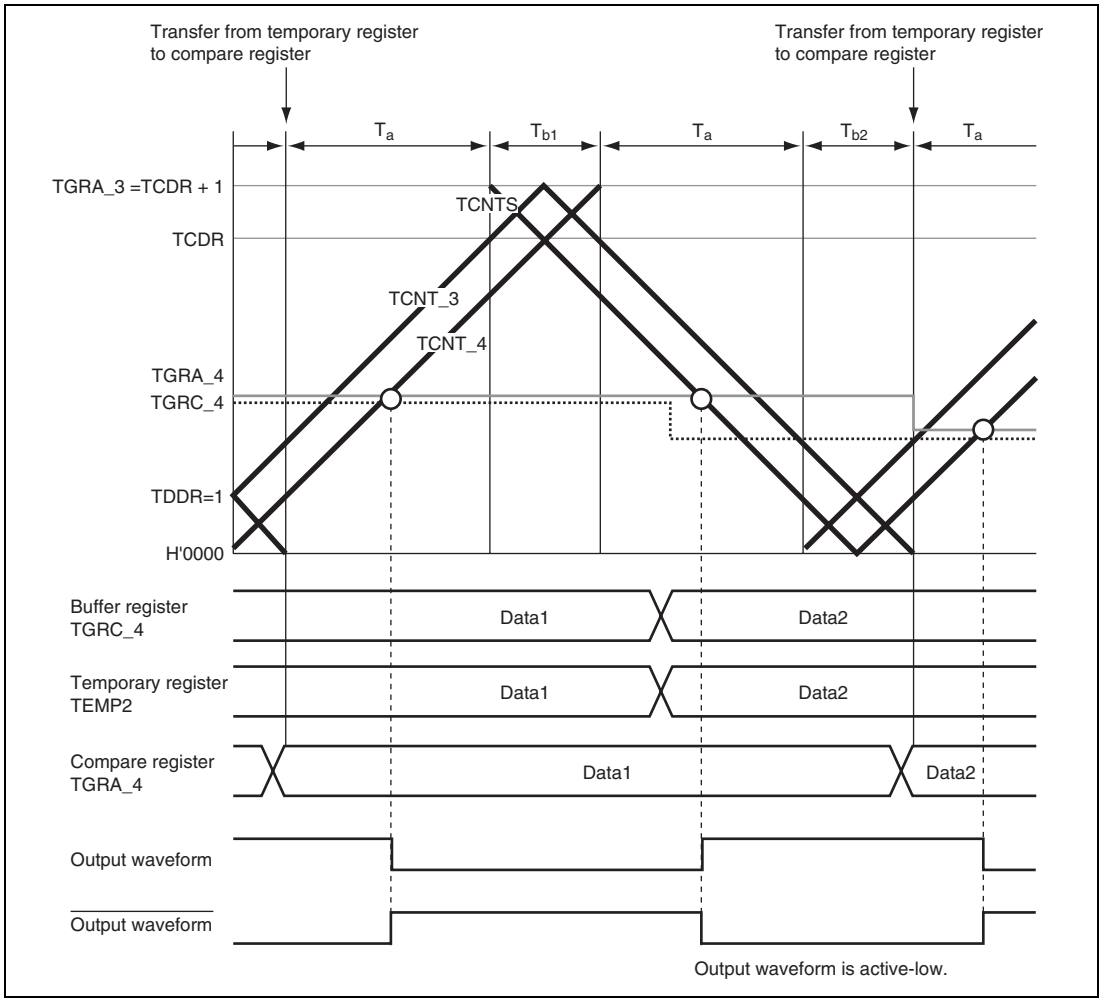


Figure 12.41 Example of Operation without Dead Time

(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 12.42 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.

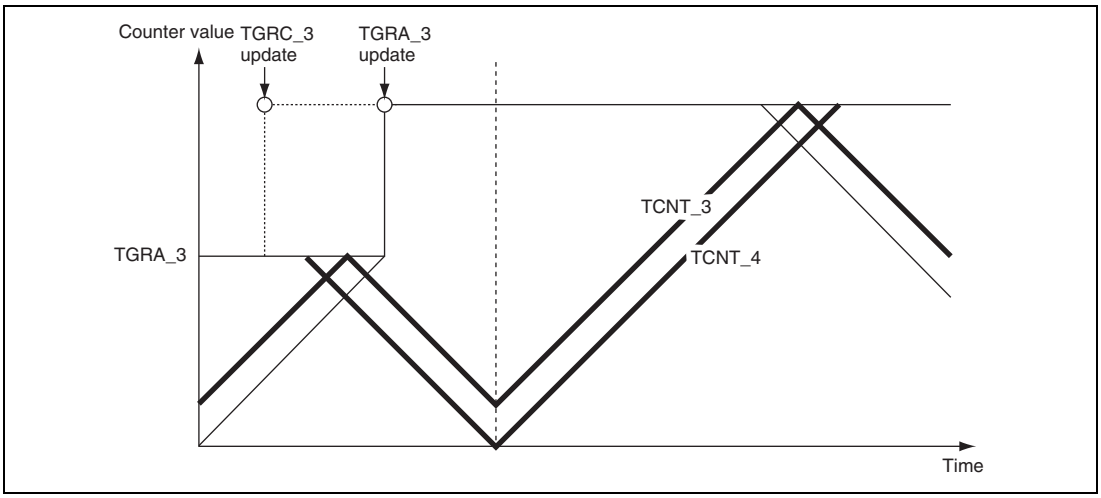


Figure 12.42 Example of PWM Cycle Updating

(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 12.43 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.

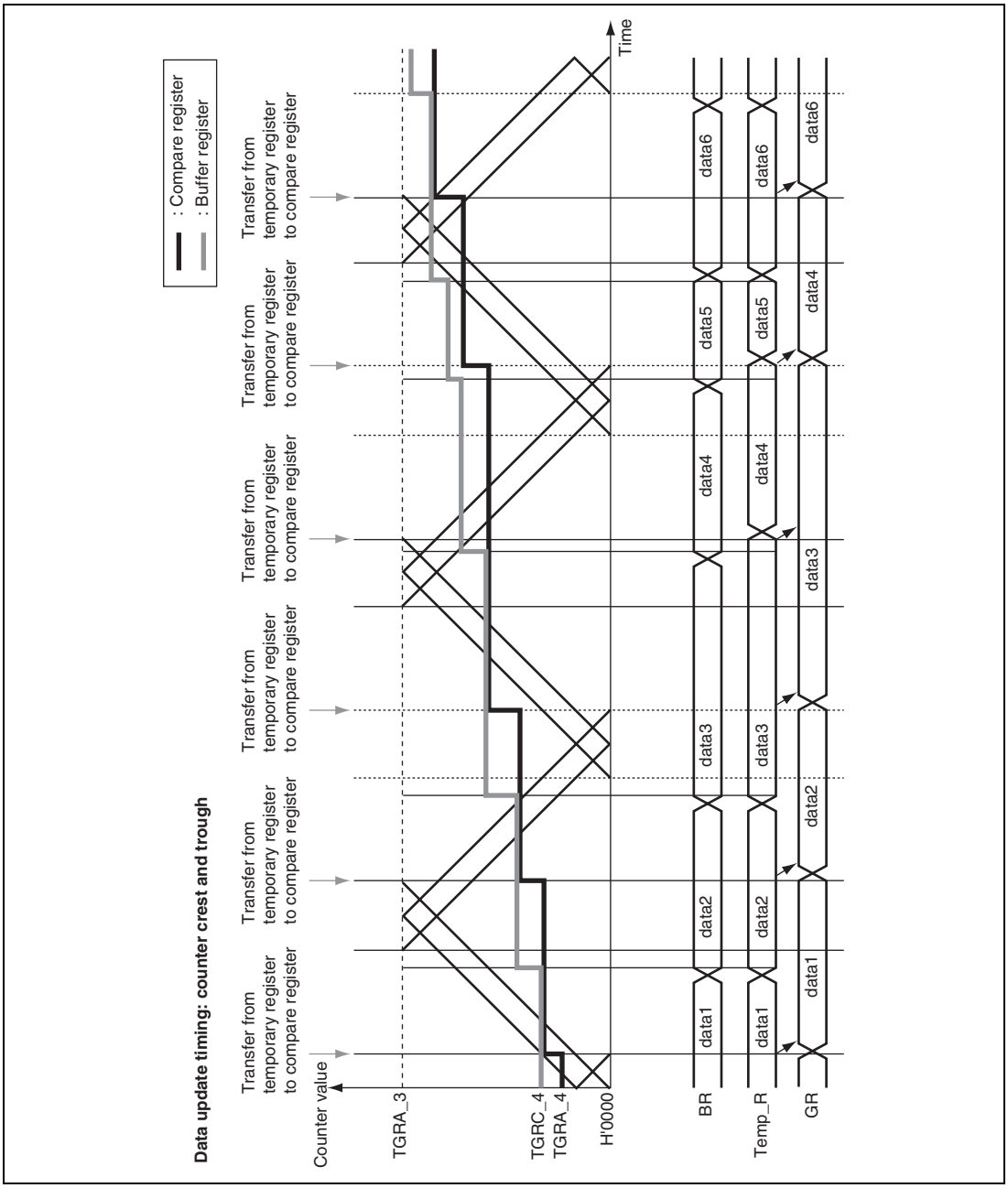


Figure 12.43 Example of Data Update in Complementary PWM Mode

(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 12.44 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 12.45.

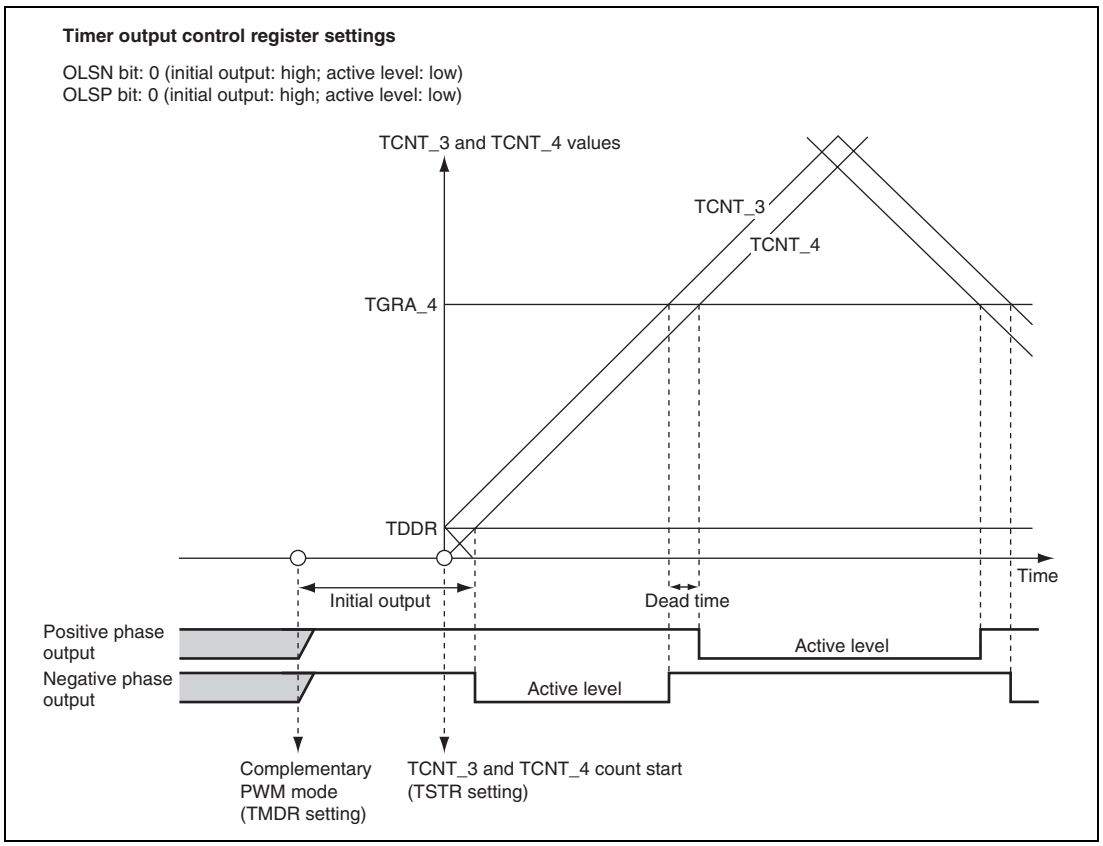


Figure 12.44 Example of Initial Output in Complementary PWM Mode (1)

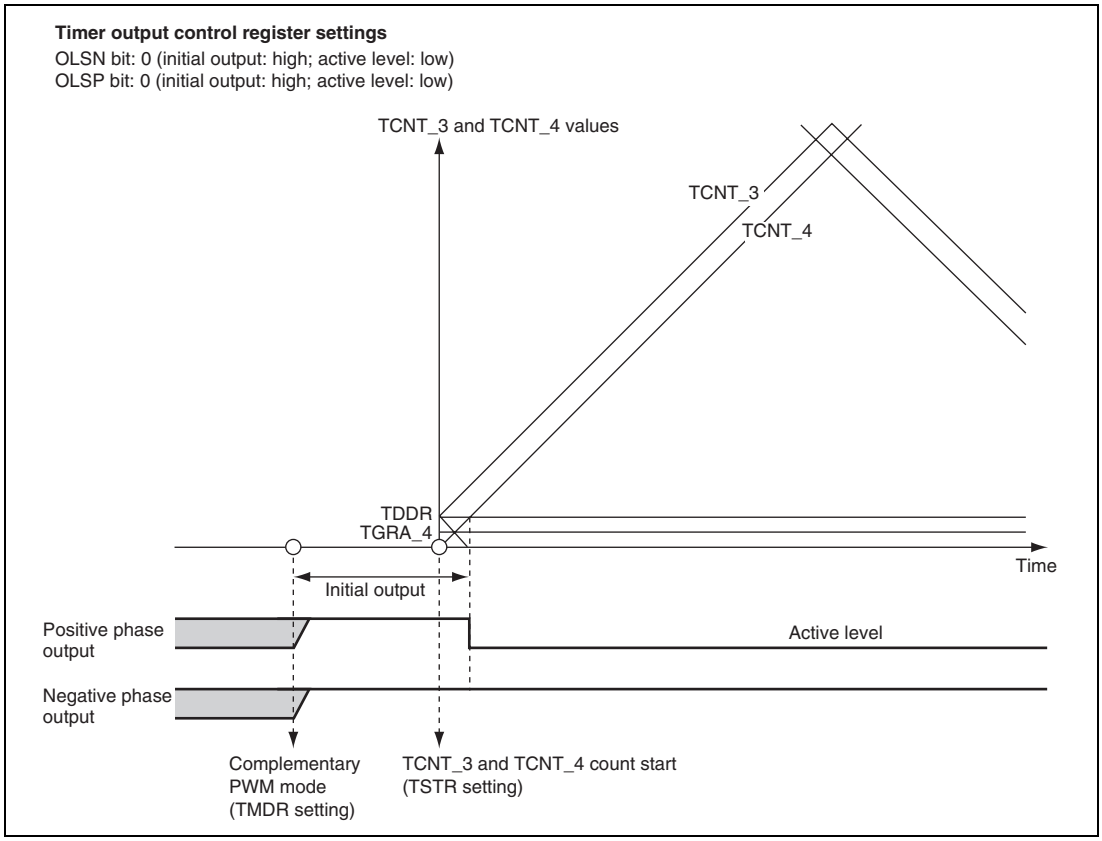


Figure 12.45 Example of Initial Output in Complementary PWM Mode (2)

(j) 10. 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 12.46 to 12.48 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 12.46.

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 12.47, 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 12.48, 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.

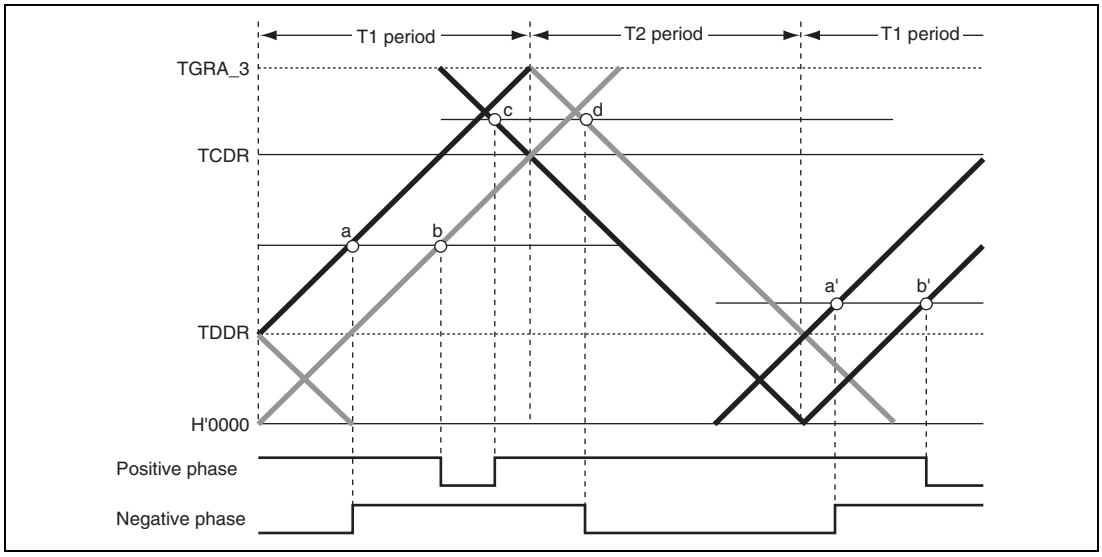


Figure 12.46 Example of Complementary PWM Mode Waveform Output (1)

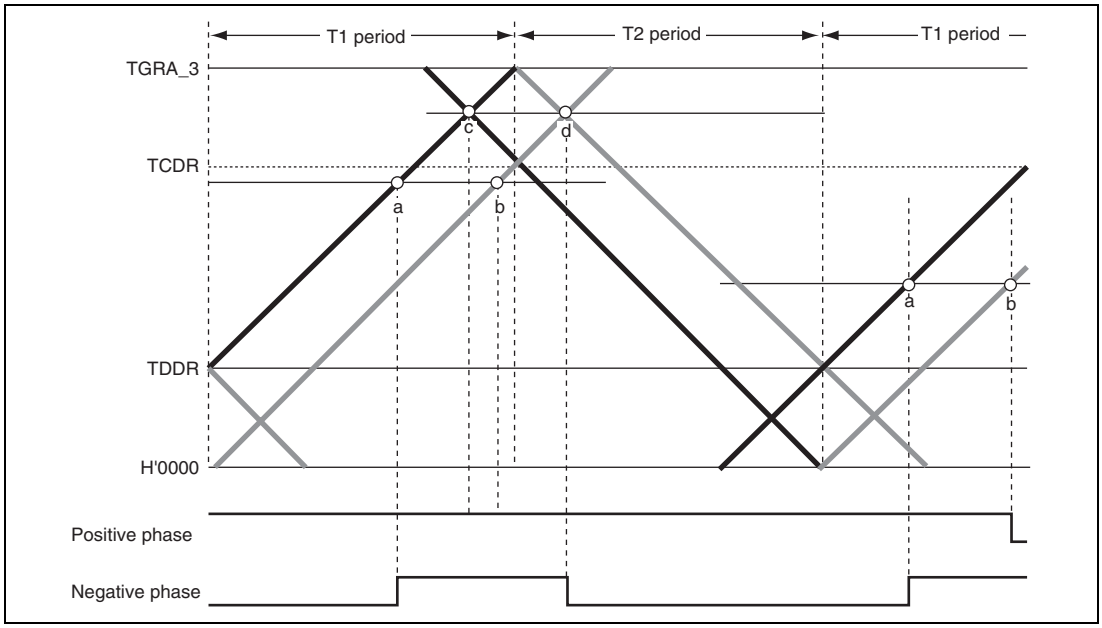


Figure 12.47 Example of Complementary PWM Mode Waveform Output (2)

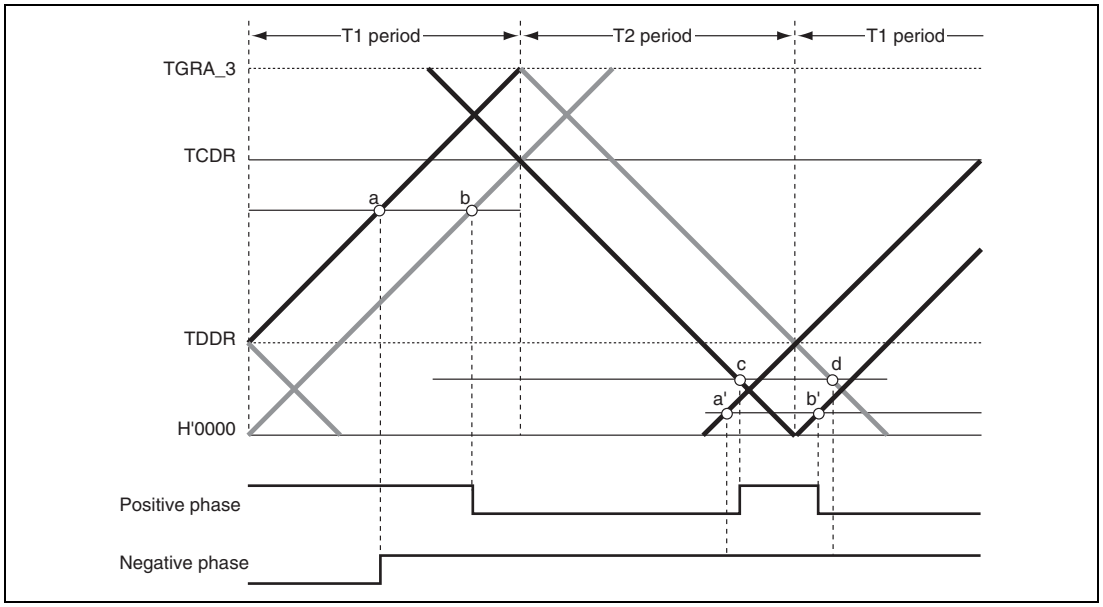


Figure 12.48 Example of Complementary PWM Mode Waveform Output (3)

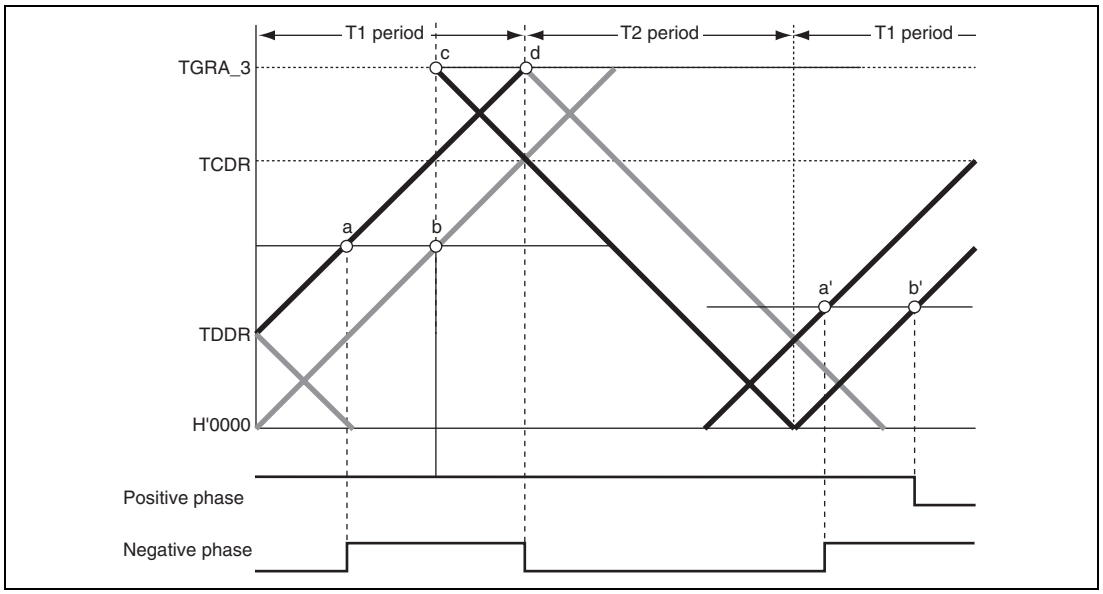


Figure 12.49 Example of Complementary PWM Mode 0% and 100% Waveform Output (1)

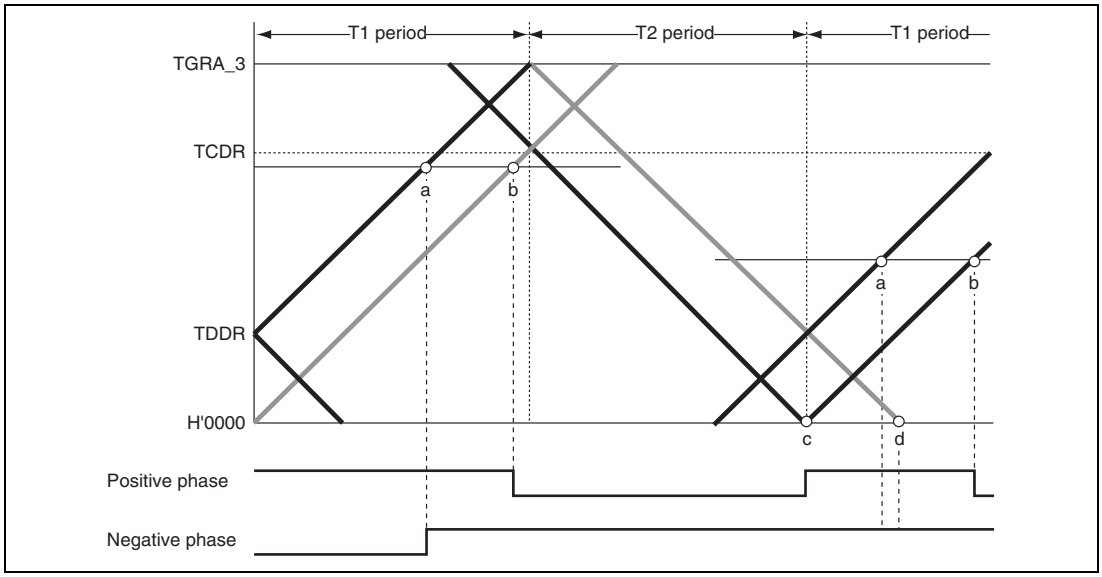


Figure 12.50 Example of Complementary PWM Mode 0% and 100% Waveform Output (2)

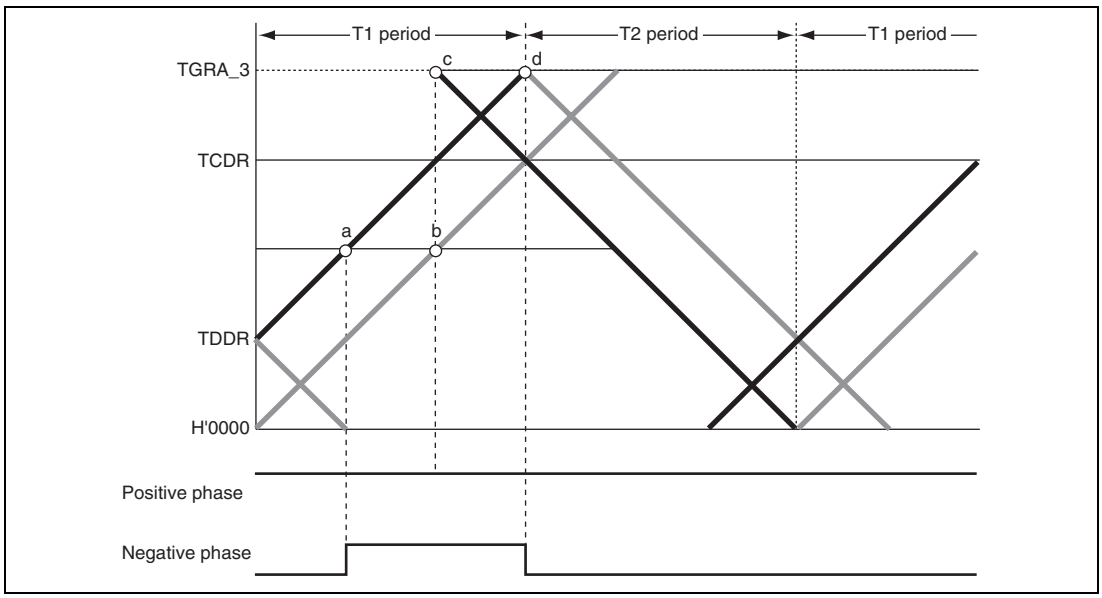


Figure 12.51 Example of Complementary PWM Mode 0% and 100% Waveform Output (3)

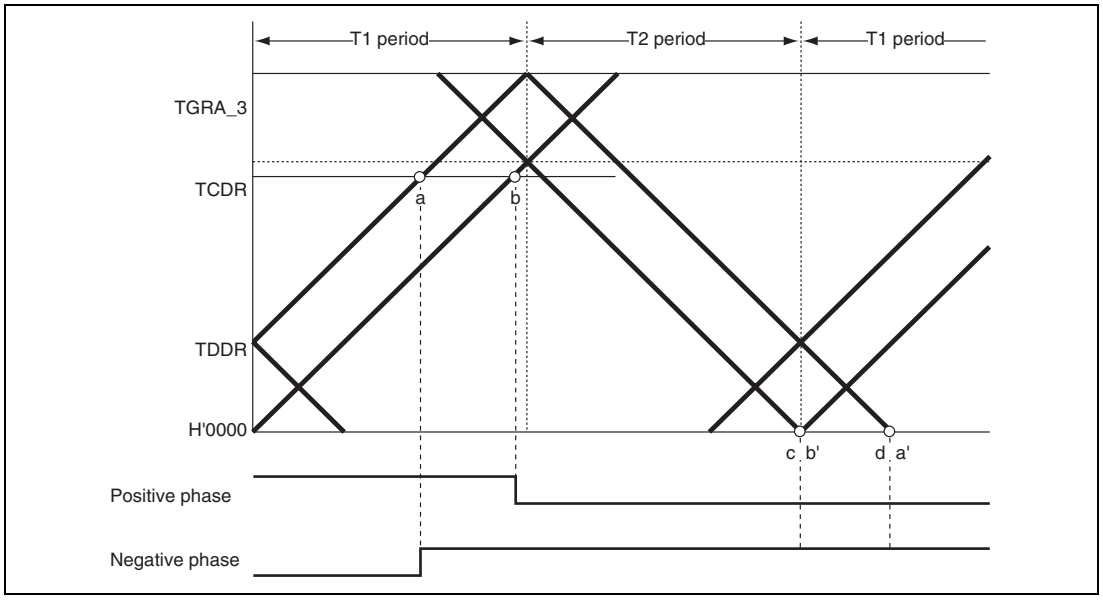


Figure 12.52 Example of Complementary PWM Mode 0% and 100% Waveform Output (4)

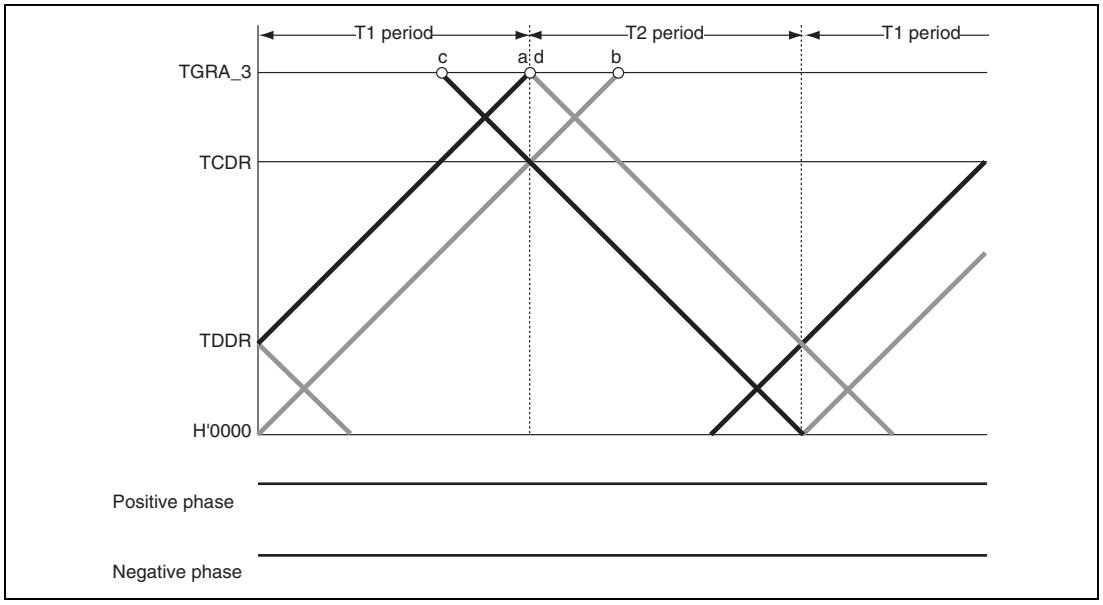


Figure 12.53 Example of Complementary PWM Mode 0% and 100% Waveform Output (5)

(k) 11. Complementary PWM Mode 0% and 100% Duty Output

In complementary PWM mode, 0% and 100% duty cycles can be output as required. Figures 12.49 to 12.53 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) 12. 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 12.54.

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.

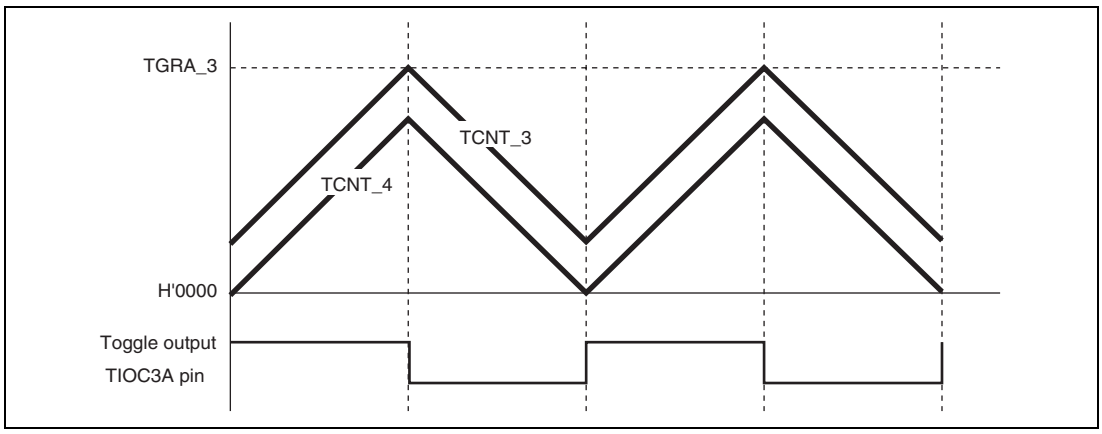


Figure 12.54 Example of Toggle Output Waveform Synchronized with PWM Output

(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 12.55 illustrates the operation.

Use of this function enables counter clearing and restarting to be performed by means of an external signal.

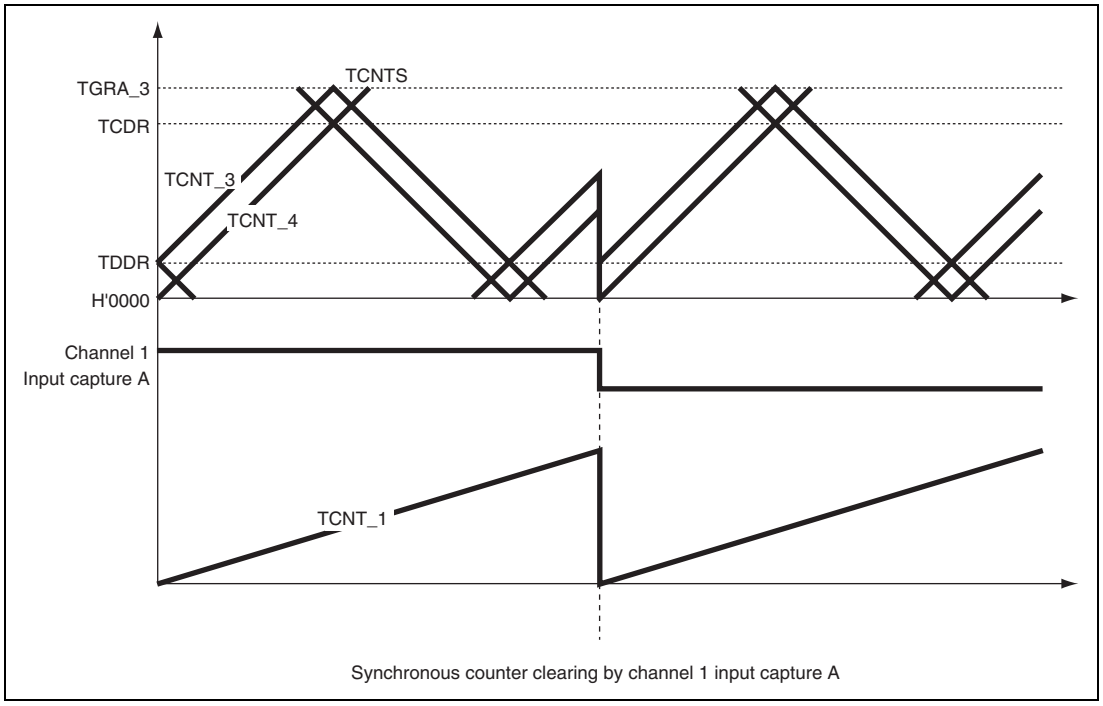


Figure 12.55 Counter Clearing Synchronized with Another Channel

(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 12.56. 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 12.56) immediately after the counters start operation, initial value output is not suppressed.

When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 12.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.

In the MTU2, synchronous clearing generated in channels 0 to 2 in the MTU2 can cause counter clearing.

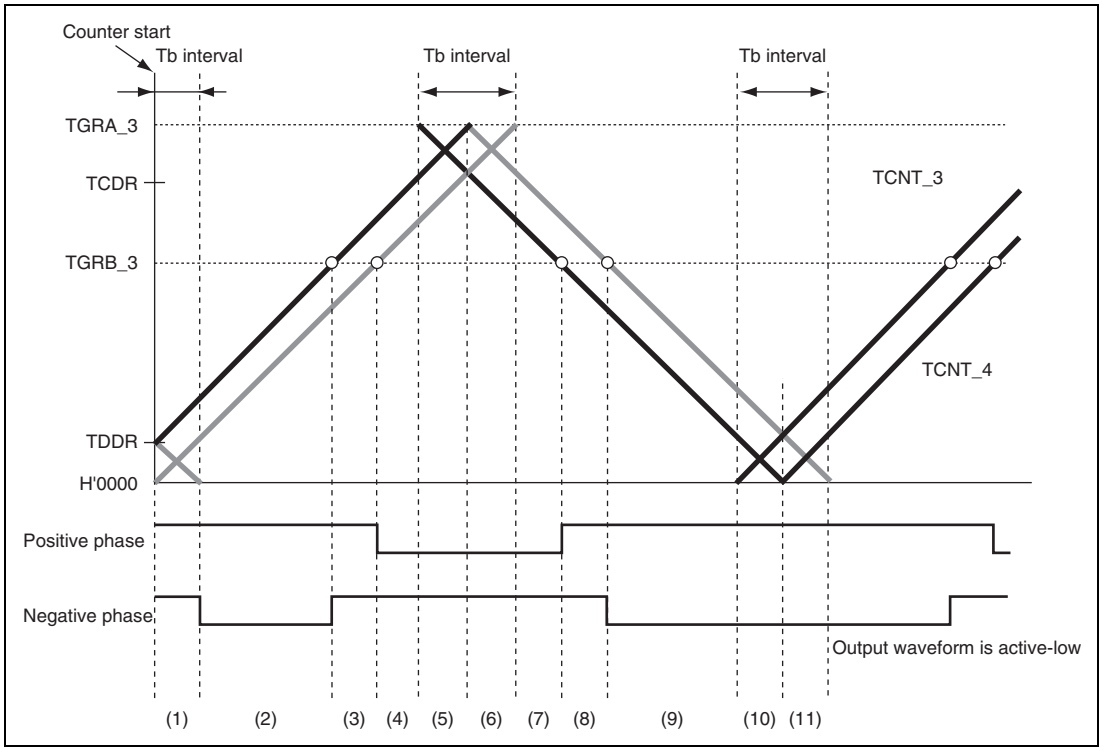


Figure 12.56 Timing for Synchronous Counter Clearing

- 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 12.57.

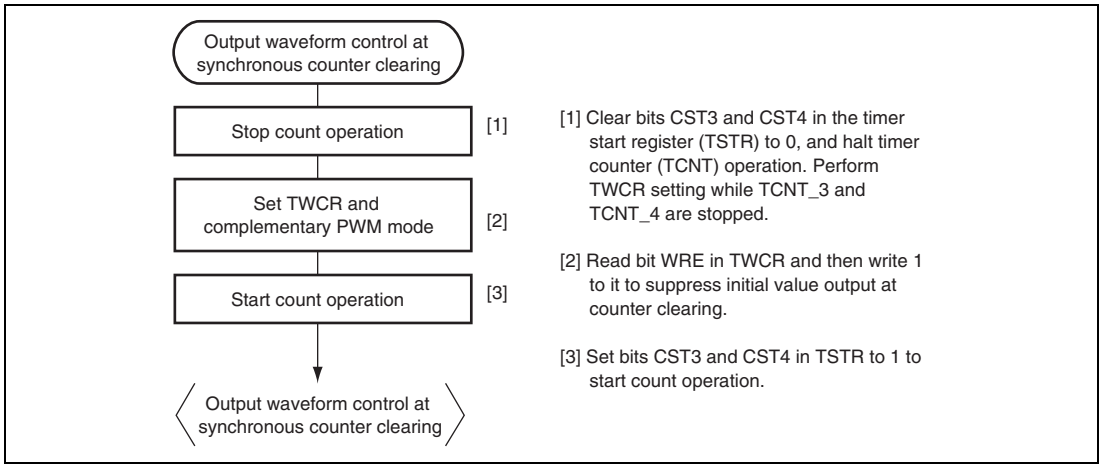


Figure 12.57 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

  Figures 12.58 to 12.61 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 12.58 to 12.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in figure 12.56, respectively.

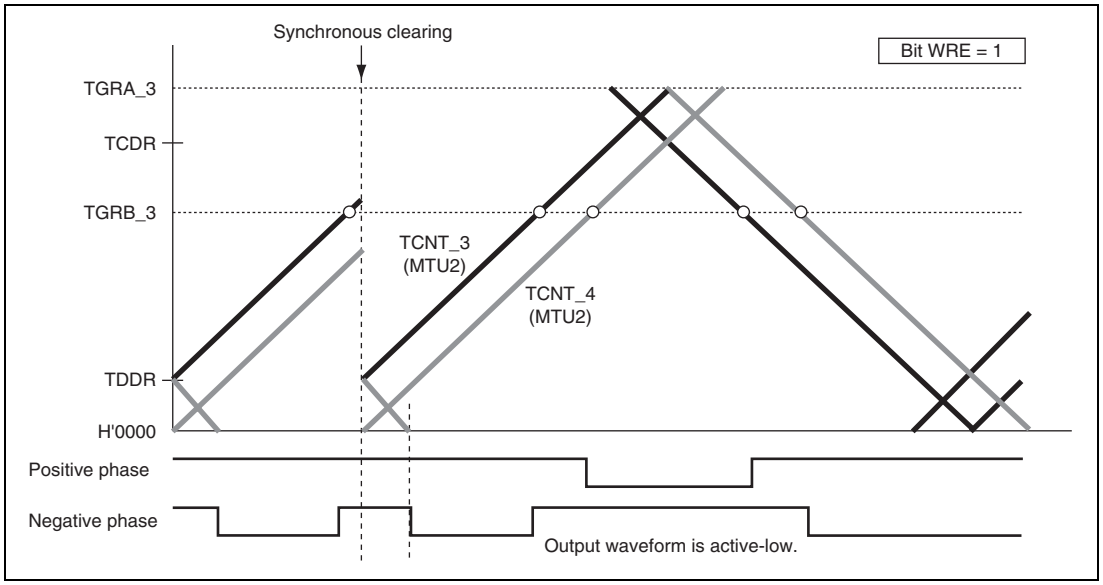


Figure 12.58 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 12.56; Bit WRE of TWCR in MTU2 is 1)

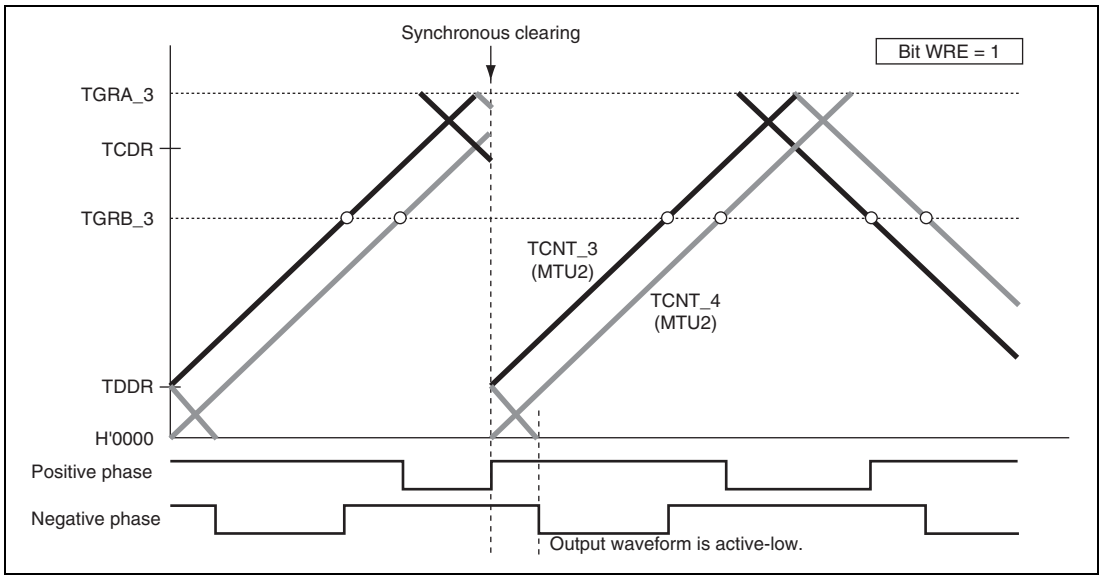


Figure 12.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 12.56; Bit WRE of TWCR in MTU2 is 1)

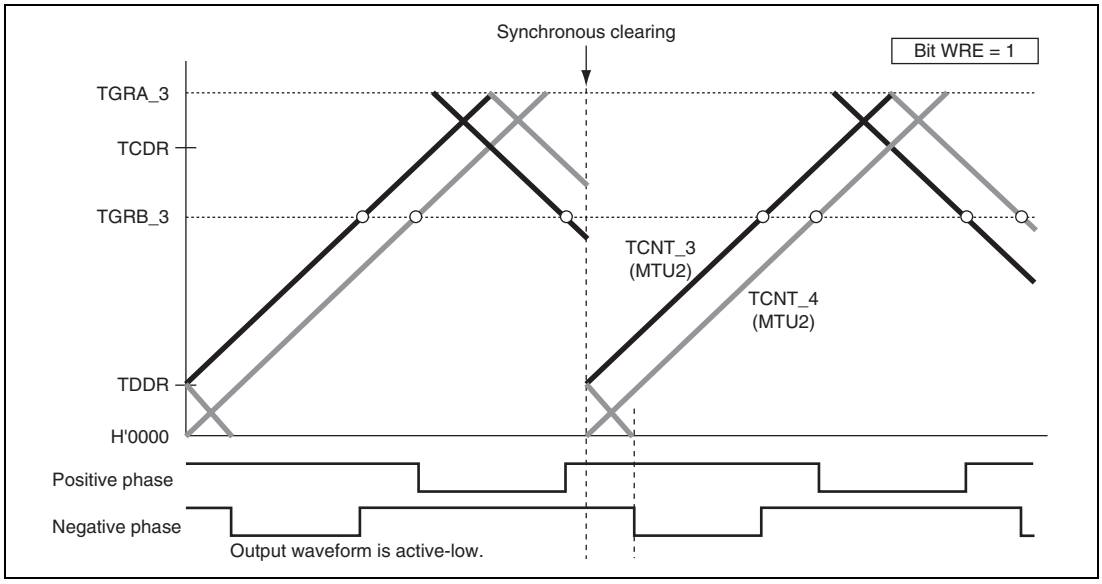


Figure 12.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 12.56; Bit WRE of TWCR is 1)

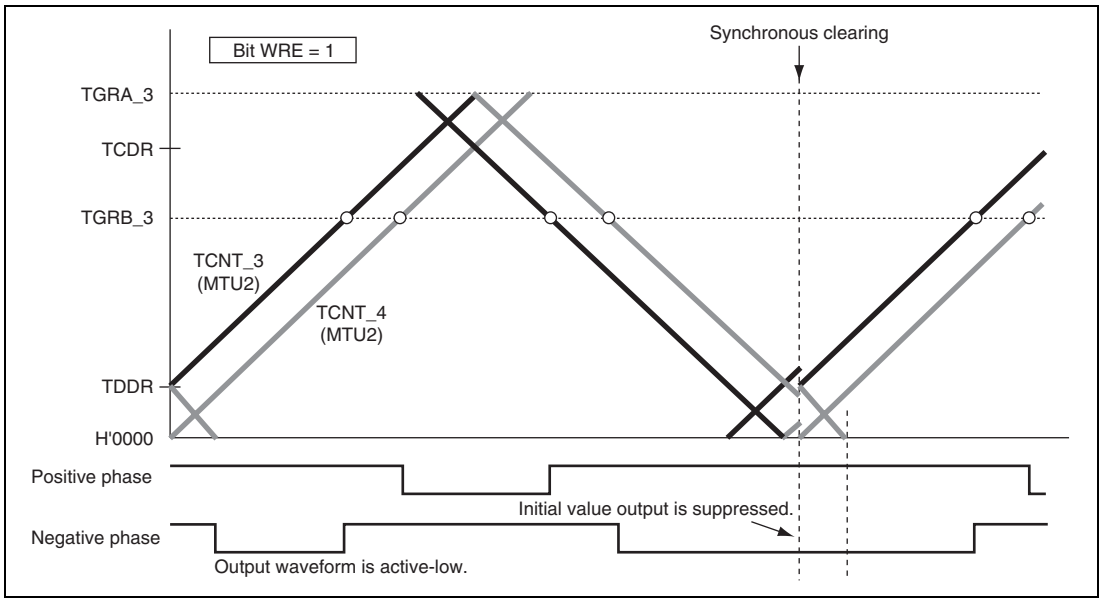


Figure 12.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 12.56; Bit WRE of TWCR is 1)

(o) 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 12.62 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).
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.

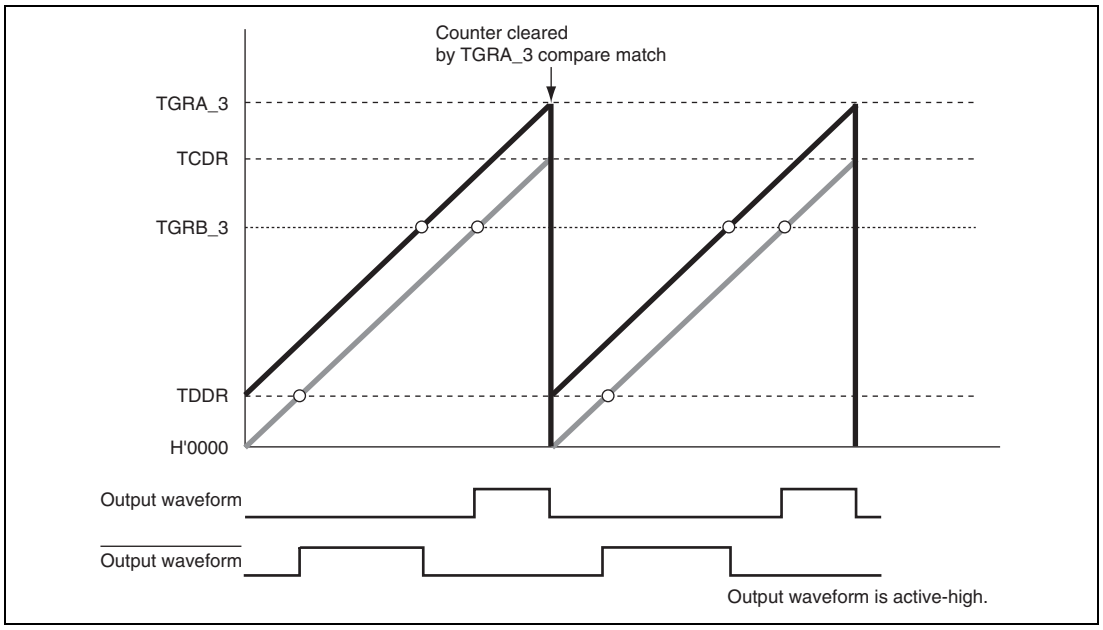


Figure 12.62 Example of Counter Clearing Operation by TGRA_3 Compare Match

(p) 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 12.63 to 12.66 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.

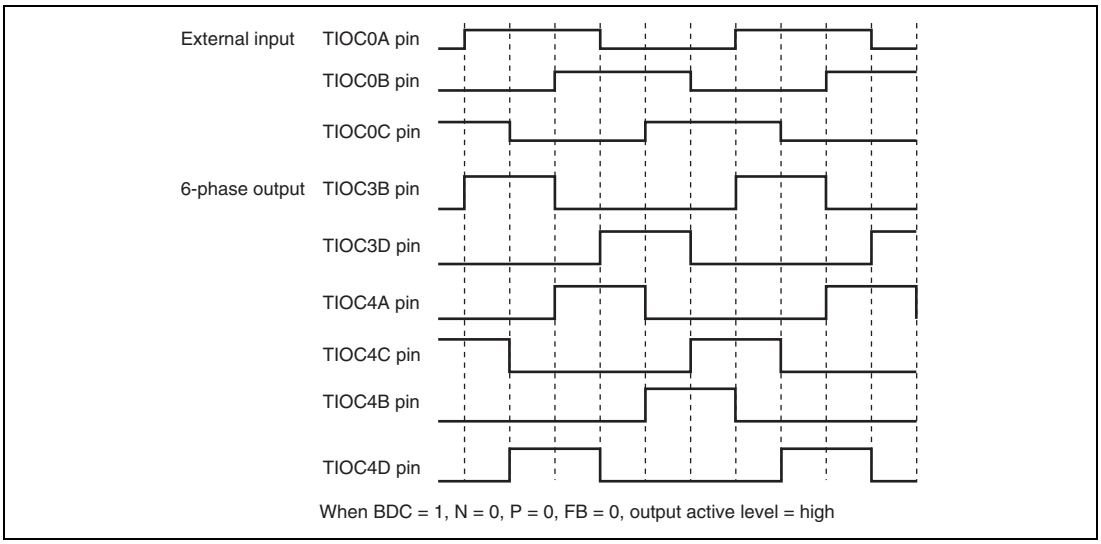


Figure 12.63 Example of Output Phase Switching by External Input (1)

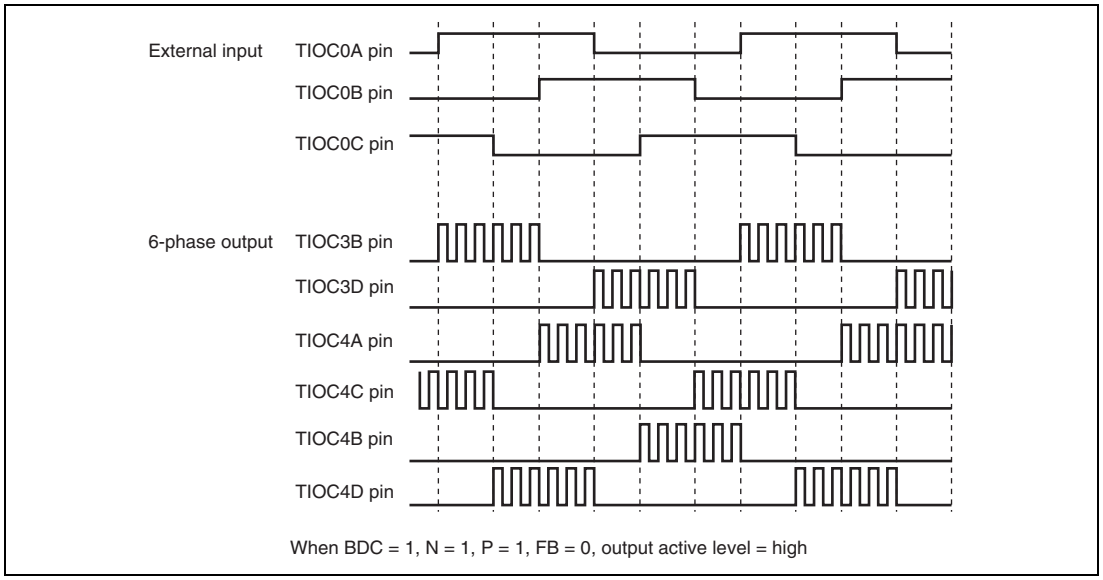


Figure 12.64 Example of Output Phase Switching by External Input (2)

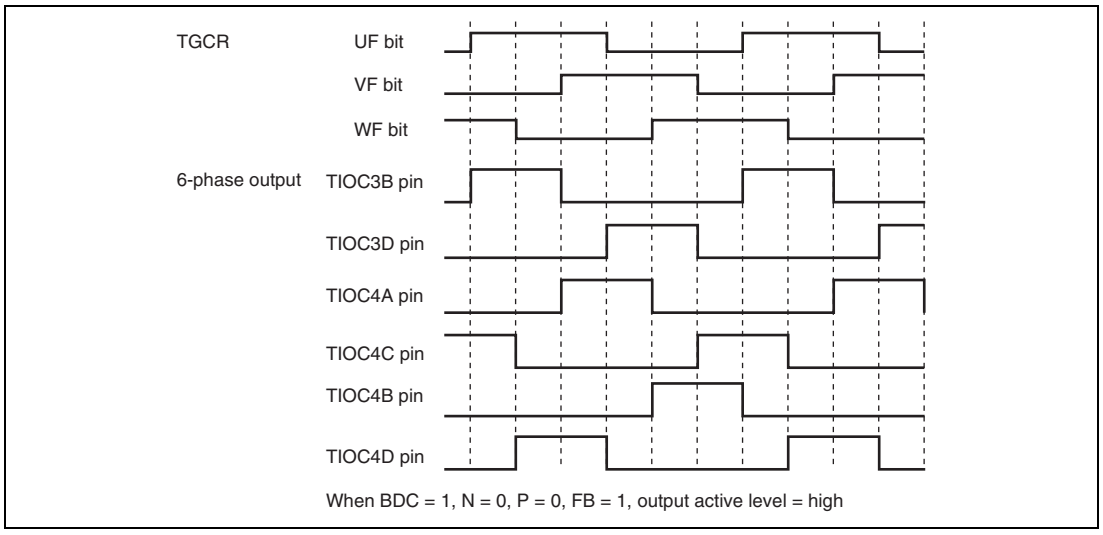


Figure 12.65 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (1)

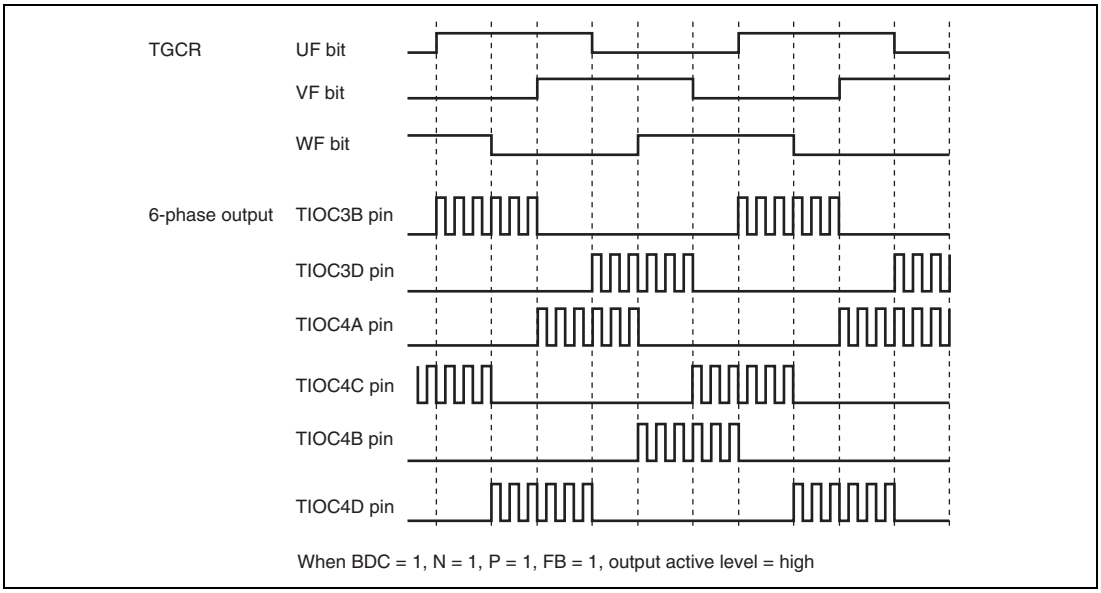


Figure 12.66 Example of Output Phase Switching by Means of UF, VF, WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using 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 12.4.9, 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 registers 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 12.67 shows an example of the interrupt skipping operation setting procedure. Figure 12.68 shows the periods during which interrupt skipping count can be changed.

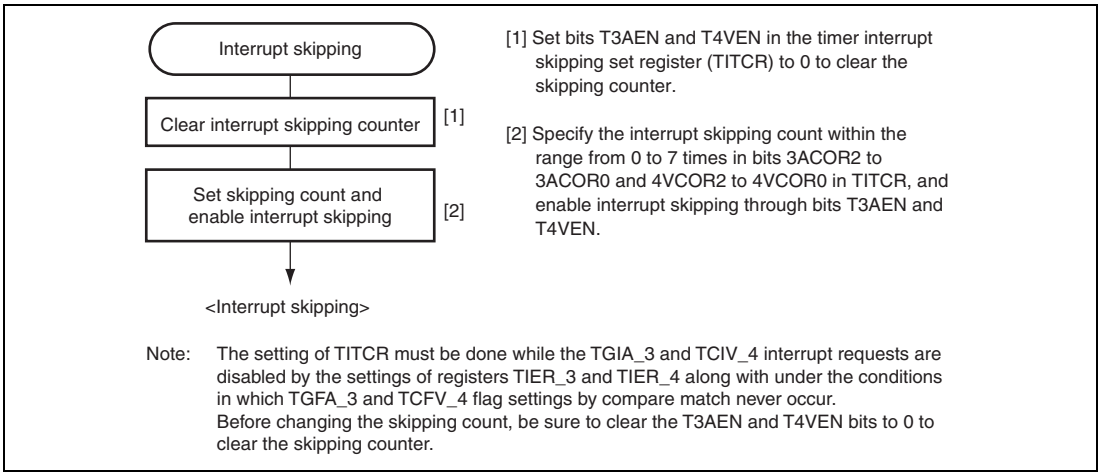


Figure 12.67 Example of Interrupt Skipping Operation Setting Procedure

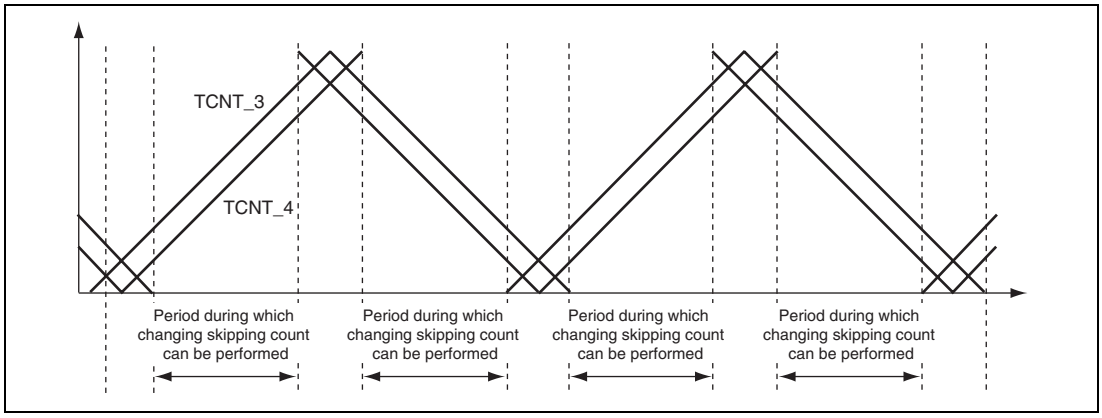


Figure 12.68 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Operation

Figure 12.69 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).

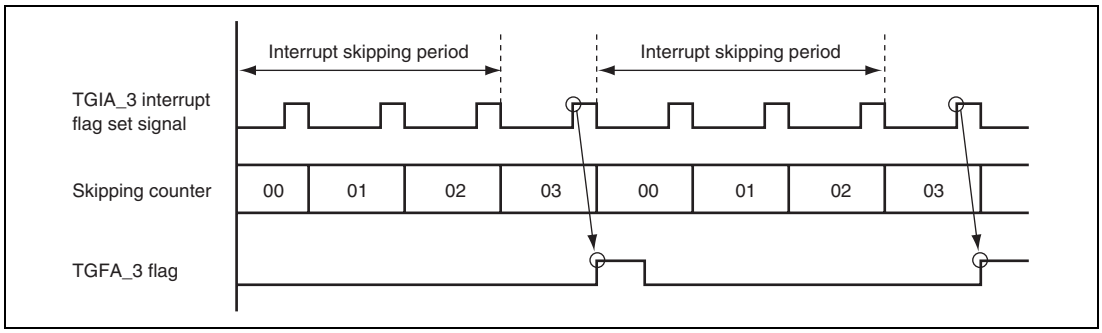


Figure 12.69 Example of Interrupt Skipping Operation

(c) Buffer Transfer Control Linked with Interrupt Skipping

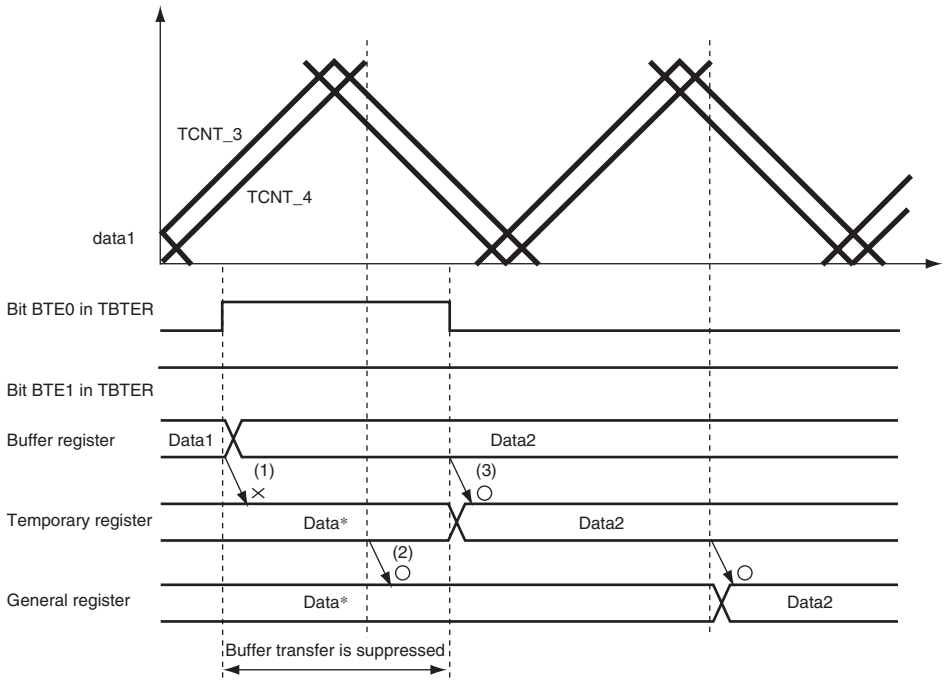
In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE1 and BTE0 bits in the timer buffer transfer set register (TBTER).

Figure 12.70 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 12.71 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.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 12.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

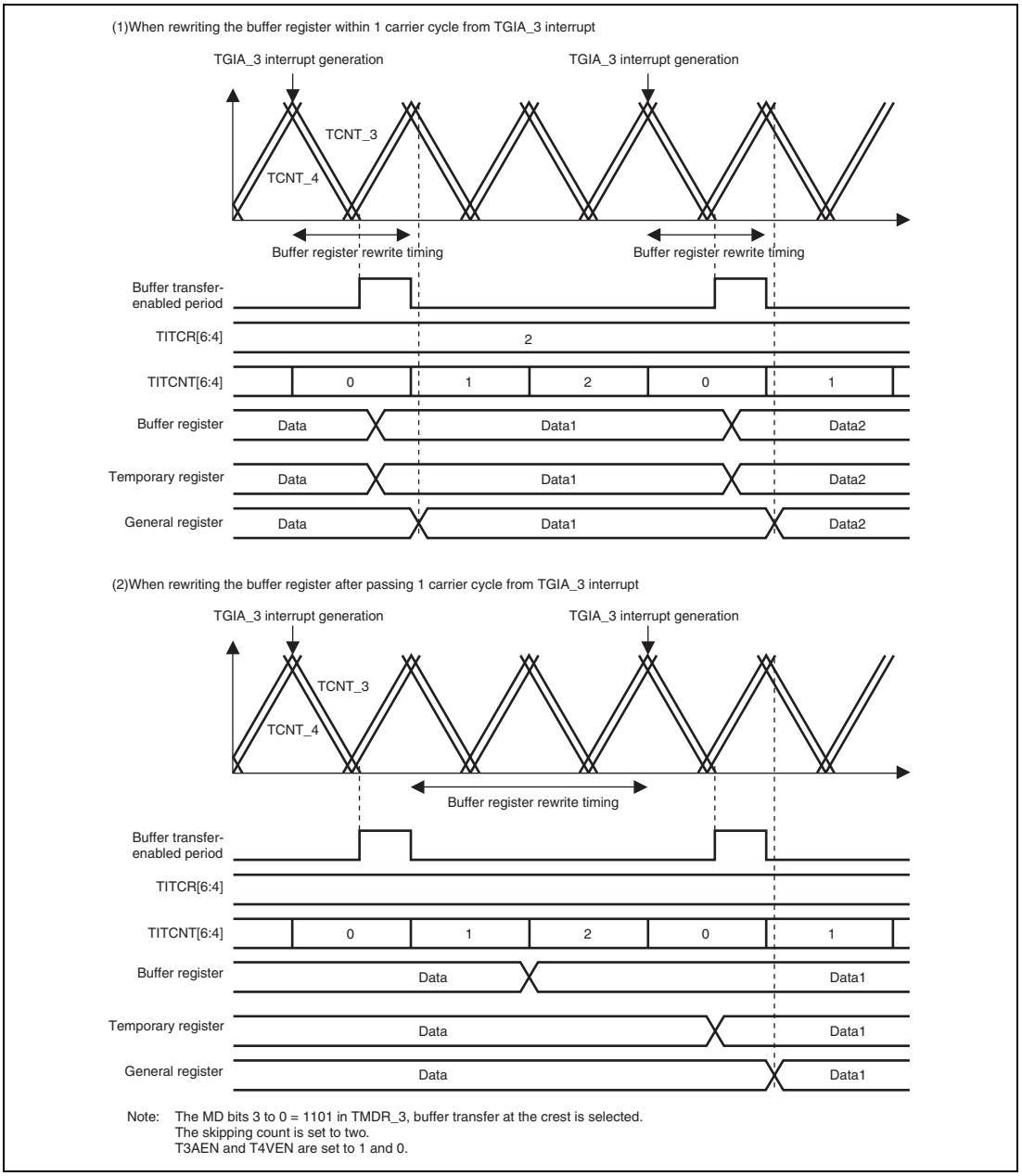
Note: This function must always be used in combination with interrupt skipping.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (3ACOR and 4VCOR) in TITCR are cleared to 0), make sure that 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 12.70 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)



Note: The MD bits 3 to 0 = 1101 in TMDR_3, buffer transfer at the crest is selected.
The skipping count is set to two.
T3AEN and T4VEN are set to 1 and 0.

Figure 12.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)

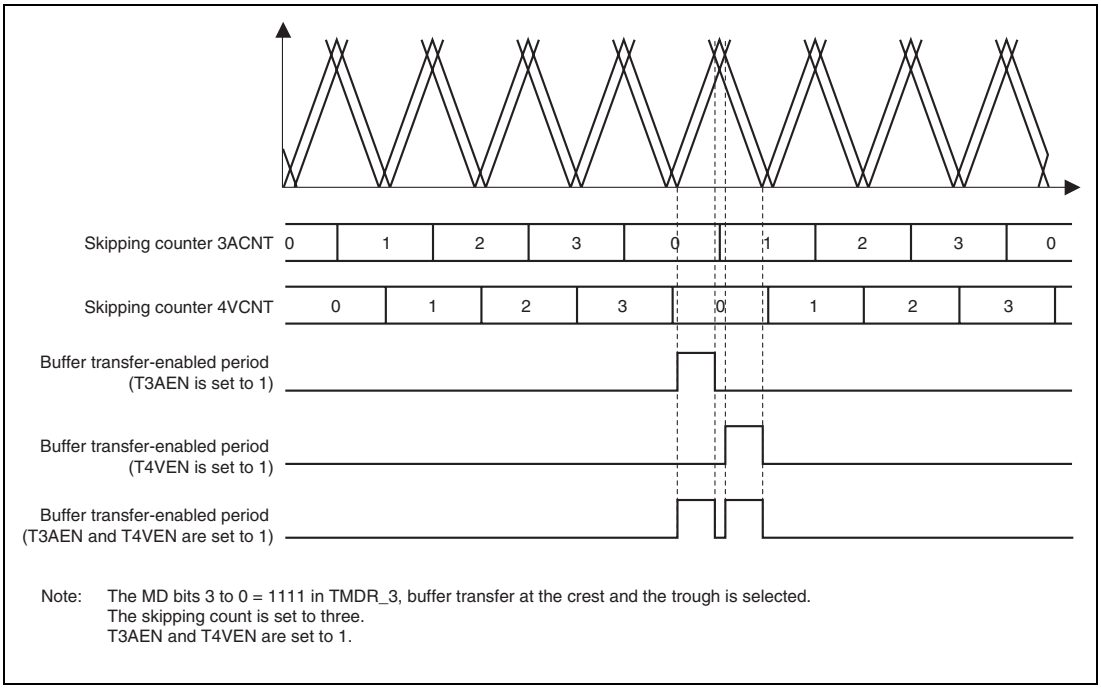


Figure 12.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period

(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.

12.4.9 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.

(a) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 12.73 shows an example of procedure for specifying the A/D converter start request delaying function.

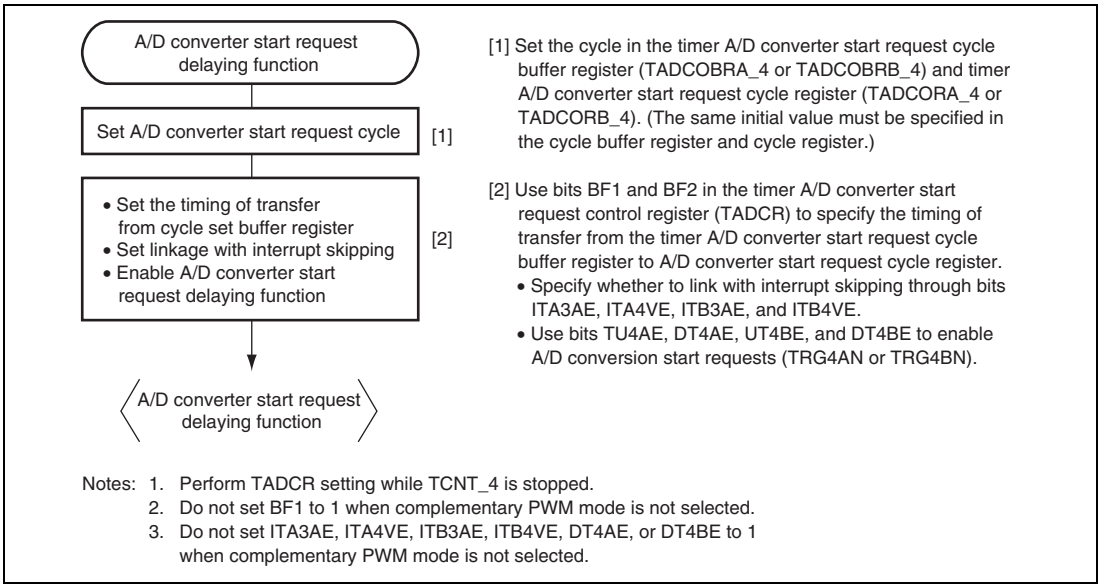


Figure 12.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

(b) Basic Operation Example of A/D Converter Start Request Delaying Function

Figure 12.74 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.

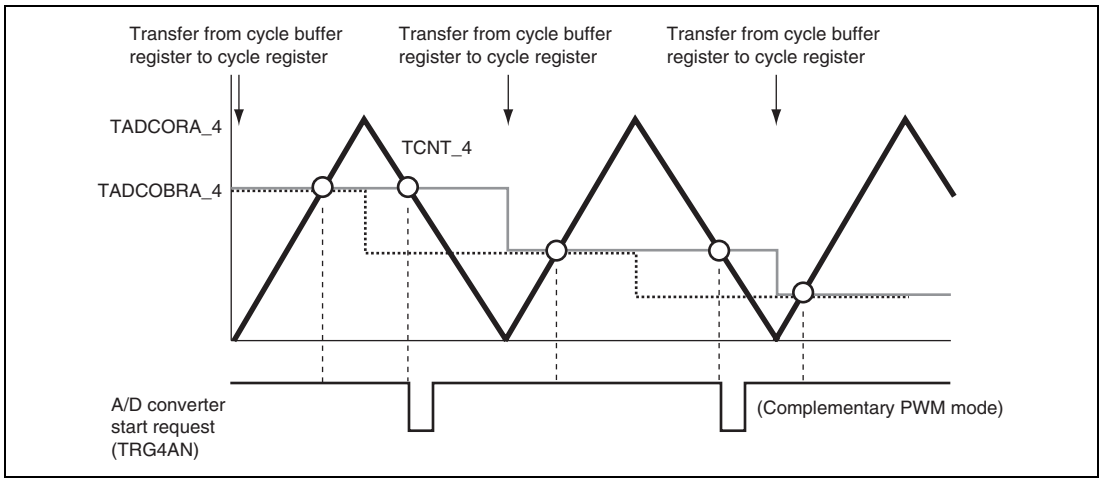


Figure 12.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

(c) 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).

(d) 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 12.75 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 12.76 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).

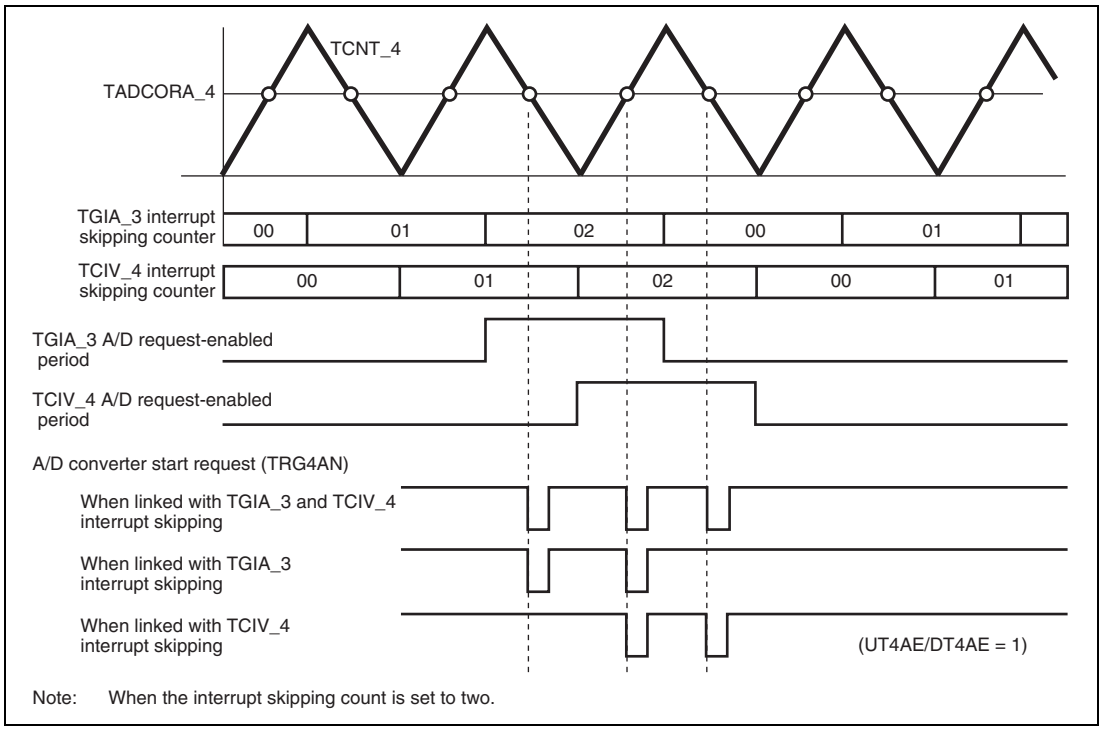


Figure 12.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

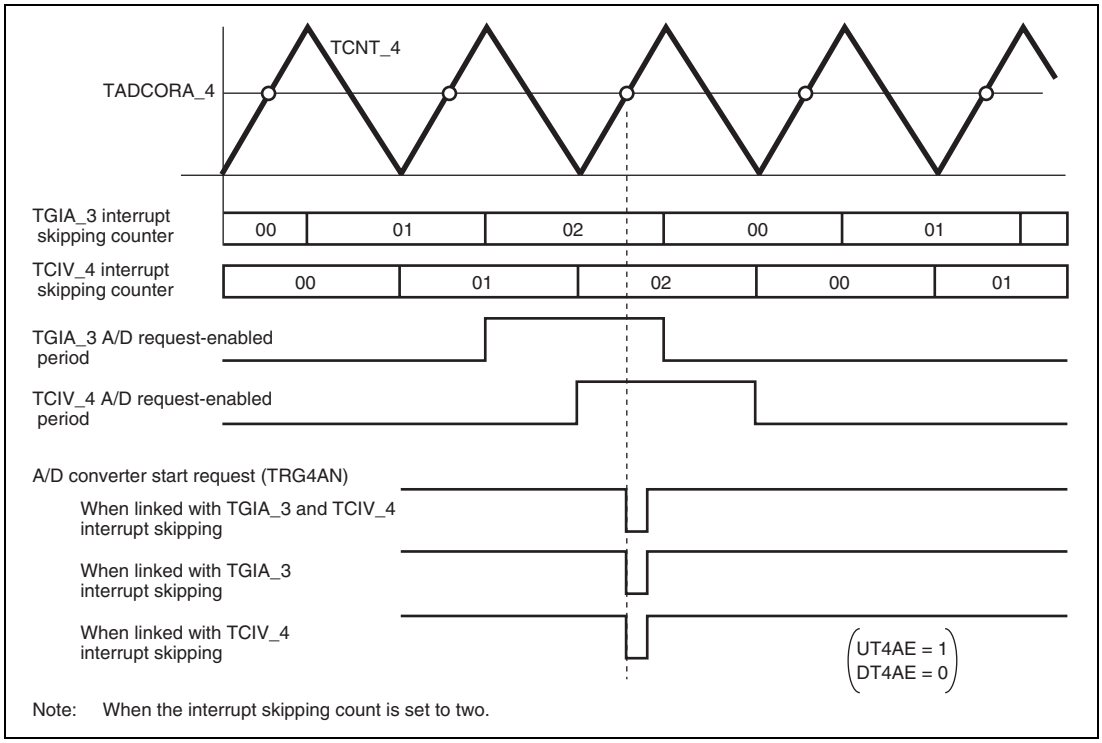


Figure 12.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping

12.4.10 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

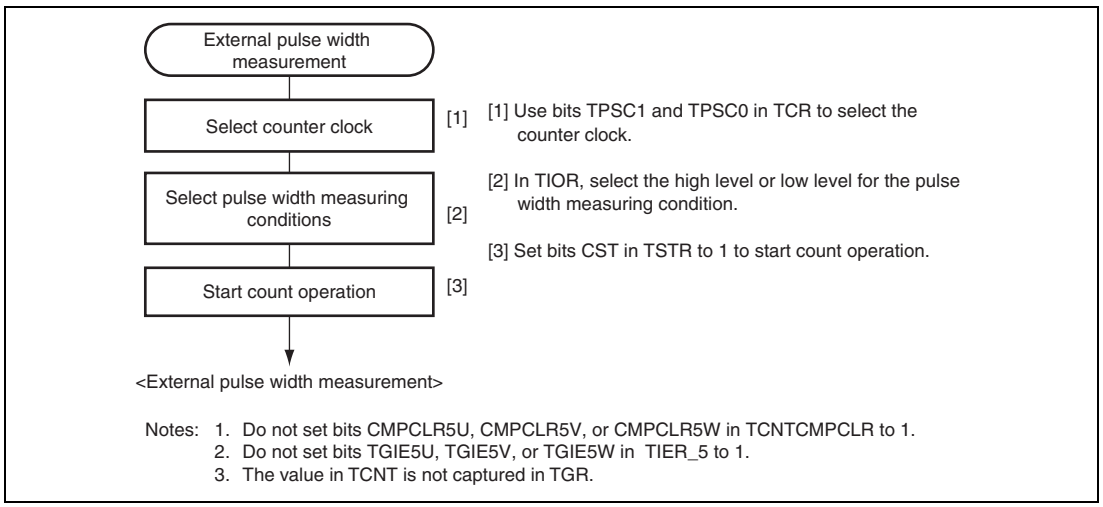


Figure 12.77 Example of External Pulse Width Measurement Setting Procedure

(2) Example of External Pulse Width Measurement

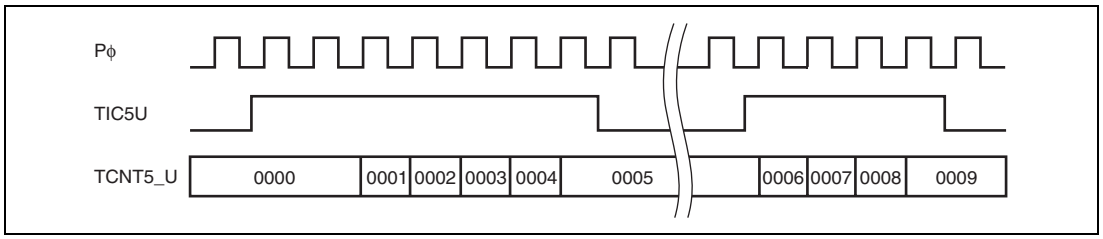


Figure 12.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)

12.4.11 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.

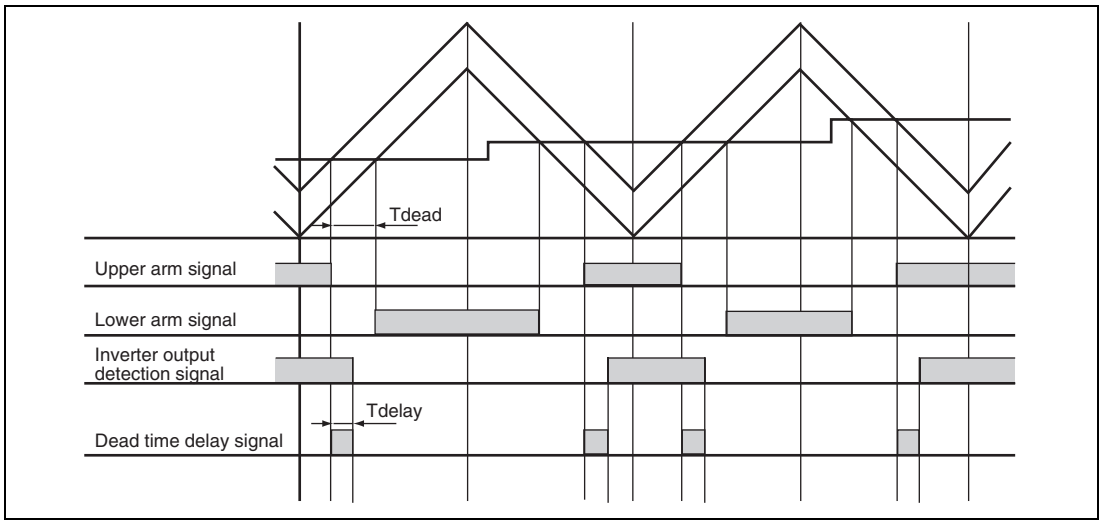


Figure 12.79 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 12.80 shows an example of dead time compensation setting procedure by using three counters in channel 5.

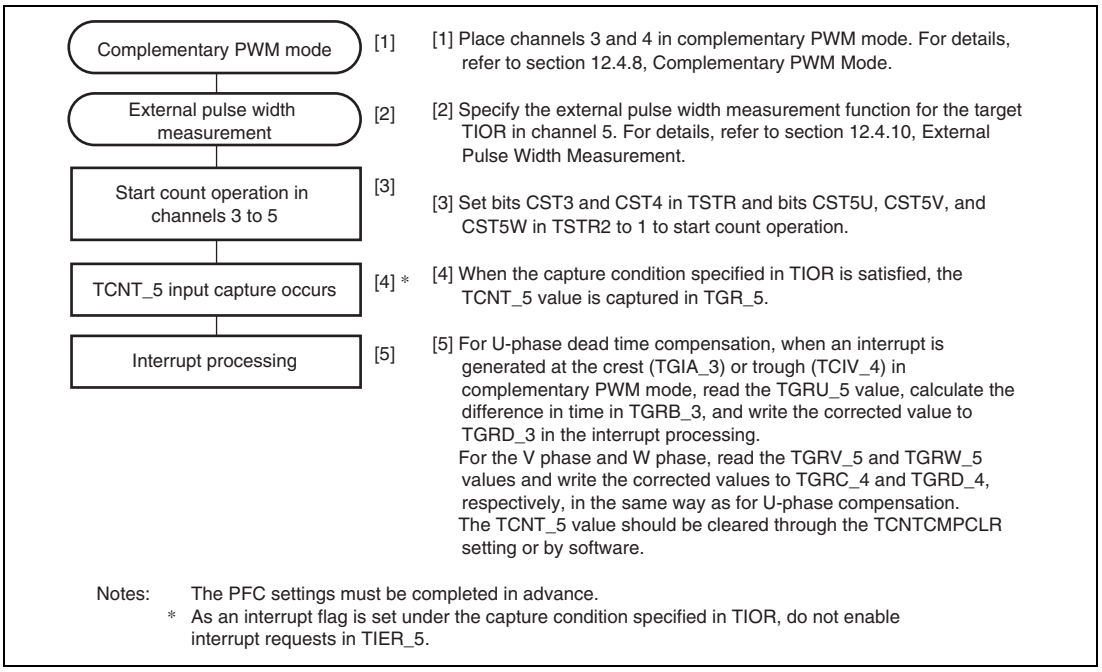


Figure 12.80 Example of Dead Time Compensation Setting Procedure

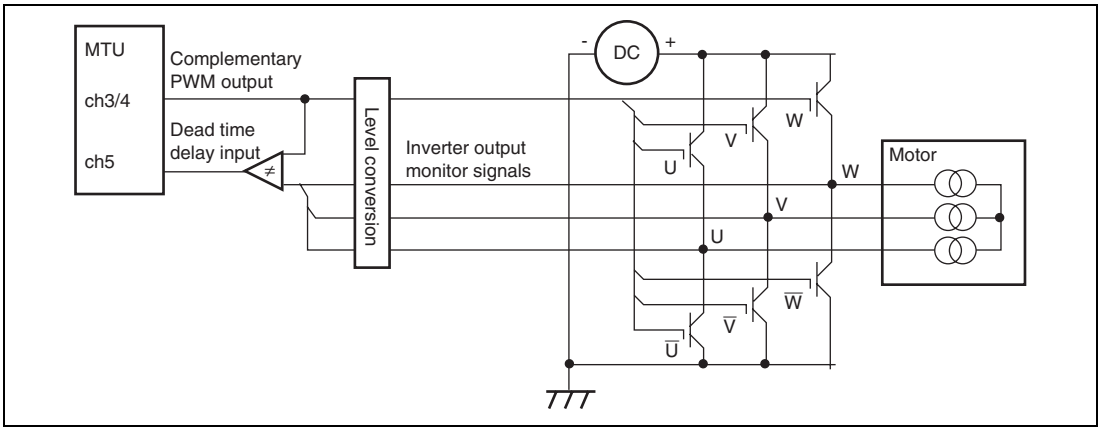


Figure 12.81 Example of Motor Control Circuit Configuration

12.4.12 TCNT Capture at Crest and/or Trough in Complementary PWM Operation

The TCNT value is captured in TGR at either the crest or trough or at both the crest and trough during complementary PWM operation. The timing for capturing in TGR can be selected by TIOR.

Figure 12.82 is an operating example in which TCNT is used as a free-running counter without being cleared, and the TCNT value is captured in TGR at the specified timing (either crest or trough, or both crest and trough).

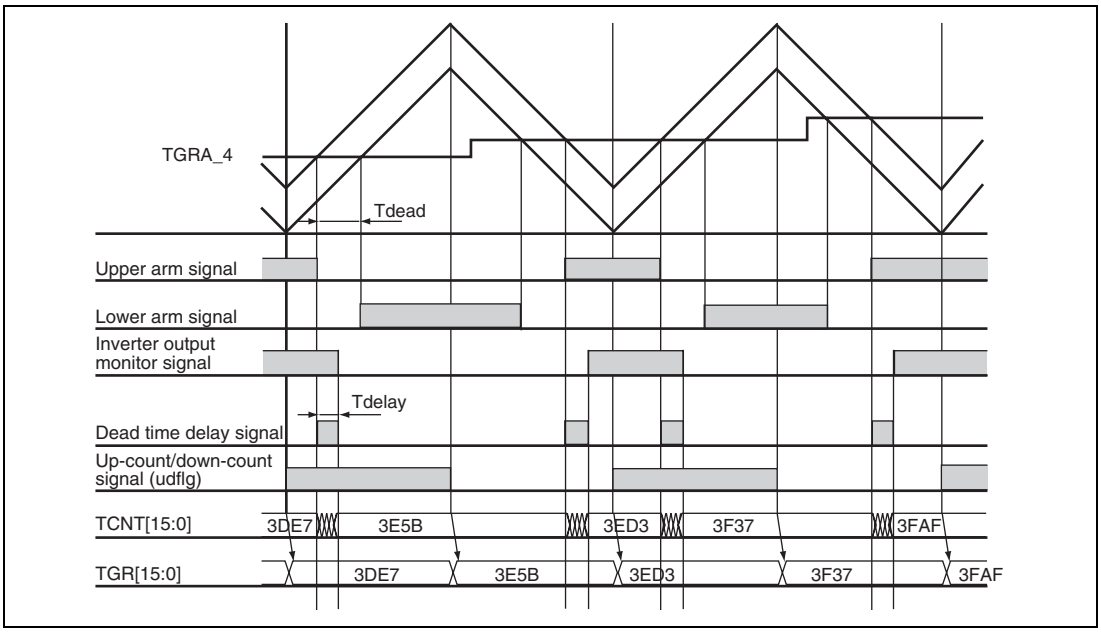


Figure 12.82 TCNT Capturing at Crest and/or Trough in Complementary PWM Operation

12.5 Interrupt Sources

12.5.1 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 12.57 lists the MTU2 interrupt sources.

Table 12.57 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	Low

Channel	Name	Interrupt Source	Interrupt Flag	DMAC Activation	Priority
3	TGIA_3	TGRA_3 input capture/compare match	TGFA_3	Possible	High
	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.

12.5.2 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 11, 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.

12.5.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU2. Table 12.58 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Converter Activation by TGRA Input Capture/Compare Match or at TCNT_4 Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in TIER_4 is set to 1, the A/D converter can be activated at the trough of TCNT_4 count (TCNT_4 = H'0000).

A/D converter start request signal TRGAN is issued to the A/D converter under either one of the following conditions.

- When the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel while the TTGE bit in TIER is set to 1
- When the TCNT_4 count reaches the trough (TCNT_4 = H'0000) during complementary PWM operation while the TTGE2 bit in TIER_4 is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Converter Activation by Compare Match between TCNT_0 and TGRE_0

The A/D converter can be activated by generating A/D converter start request signal TRG0N when a compare match occurs between TCNT_0 and TGRE_0 in channel 0.

When the TGFE flag in TSR2_0 is set to 1 by the occurrence of a compare match between TCNT_0 and TGRE_0 in channel 0 while the TTGE2 bit in TIER2_0 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TGR0N from the MTU2 is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Converter Activation by A/D Converter Start Request Delaying Function

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 12.4.9, 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 12.58 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

12.6 Operation Timing

12.6.1 Input/Output Timing

(1) TCNT Count Timing

Figures 12.83 and 12.84 show TCNT count timing in internal clock operation, and figure 12.85 shows TCNT count timing in external clock operation (normal mode), and figure 12.86 shows TCNT count timing in external clock operation (phase counting mode).

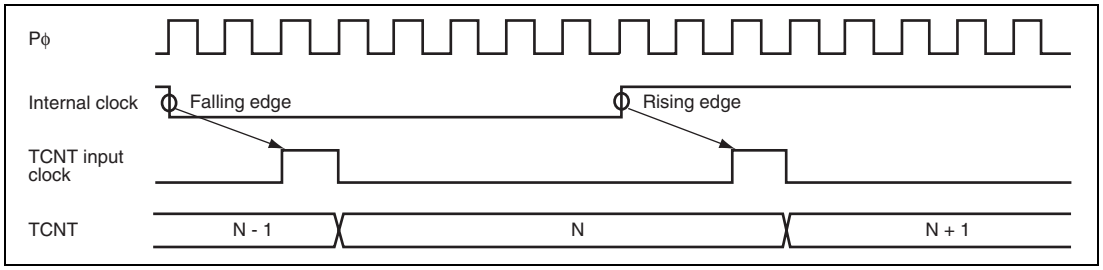


Figure 12.83 Count Timing in Internal Clock Operation (Channels 0 to 4)

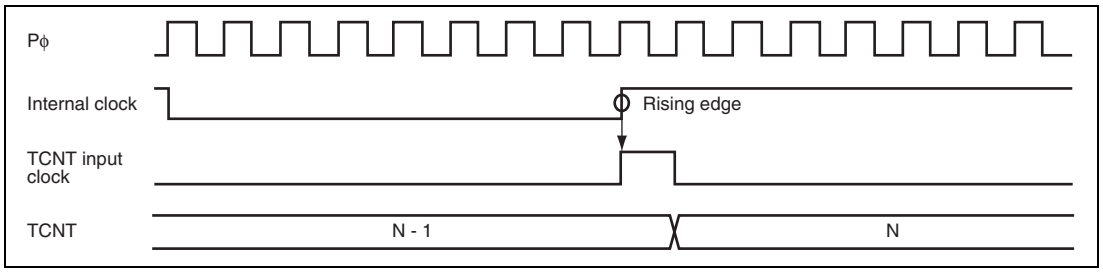


Figure 12.84 Count Timing in Internal Clock Operation (Channel 5)

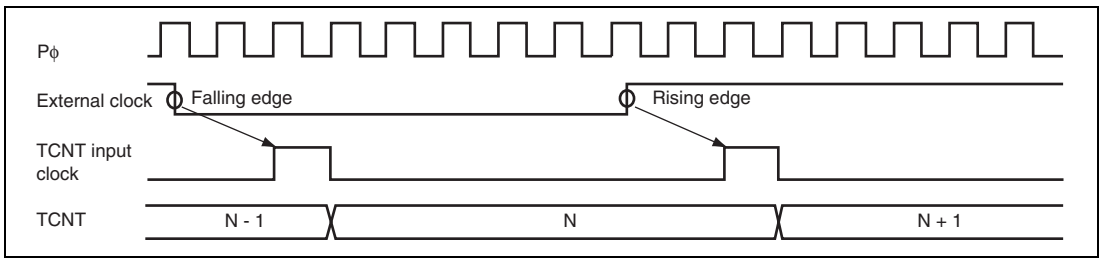


Figure 12.85 Count Timing in External Clock Operation (Channels 0 to 4)

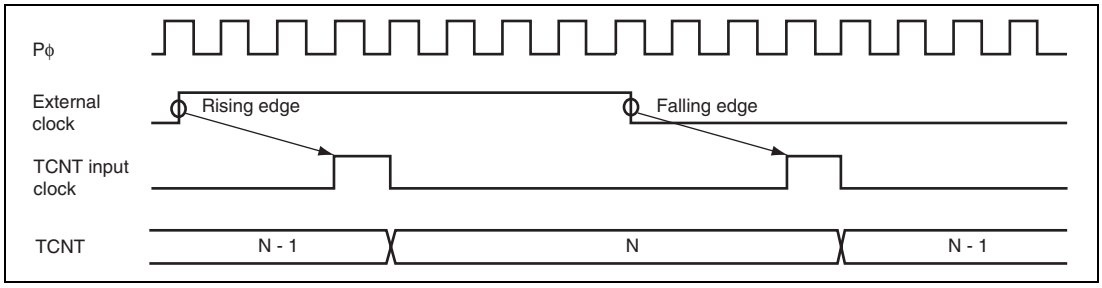


Figure 12.86 Count Timing in External Clock Operation (Phase Counting Mode)

(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 12.87 shows output compare output timing (normal mode and PWM mode) and figure 12.88 shows output compare output timing (complementary PWM mode and reset synchronous PWM mode).

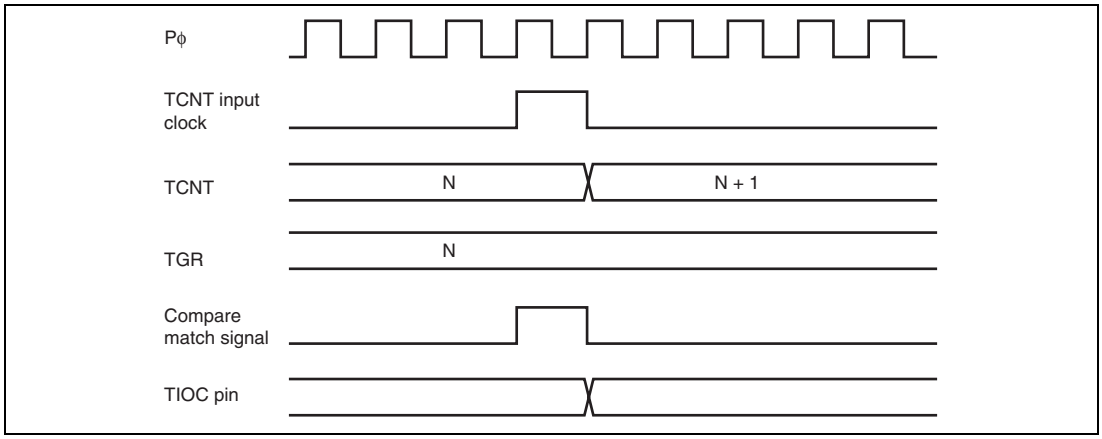


Figure 12.87 Output Compare Output Timing (Normal Mode/PWM Mode)

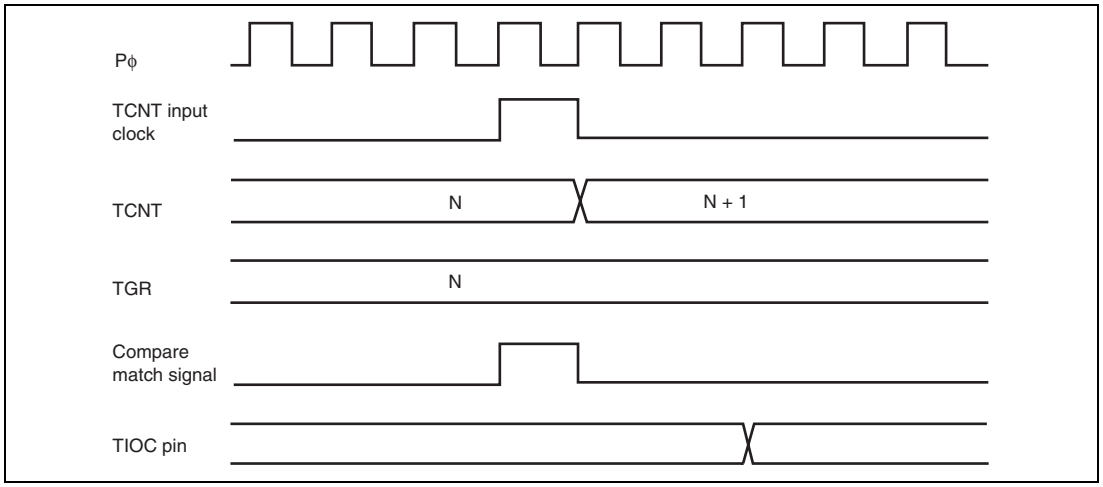


Figure 12.88 Output Compare Output Timing (Complementary PWM Mode/Reset Synchronous PWM Mode)

(3) Input Capture Signal Timing

Figure 12.89 shows input capture signal timing.

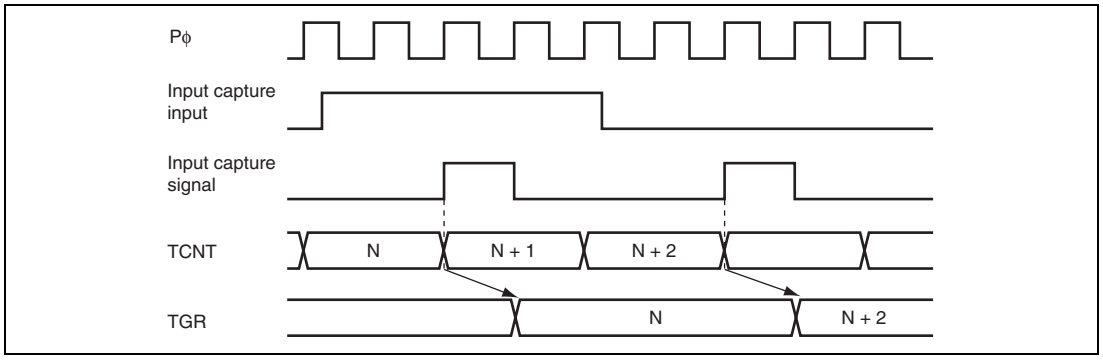


Figure 12.89 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figures 12.90 and 12.91 show the timing when counter clearing on compare match is specified, and figure 12.92 shows the timing when counter clearing on input capture is specified.

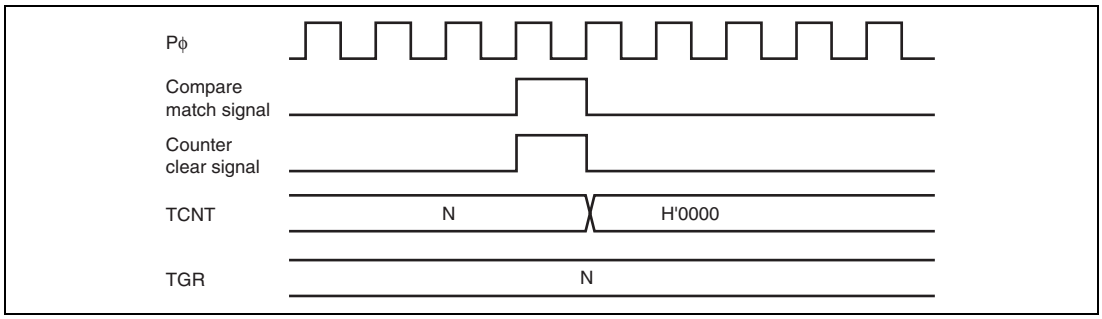


Figure 12.90 Counter Clear Timing (Compare Match) (Channel 0 to Channel 4)

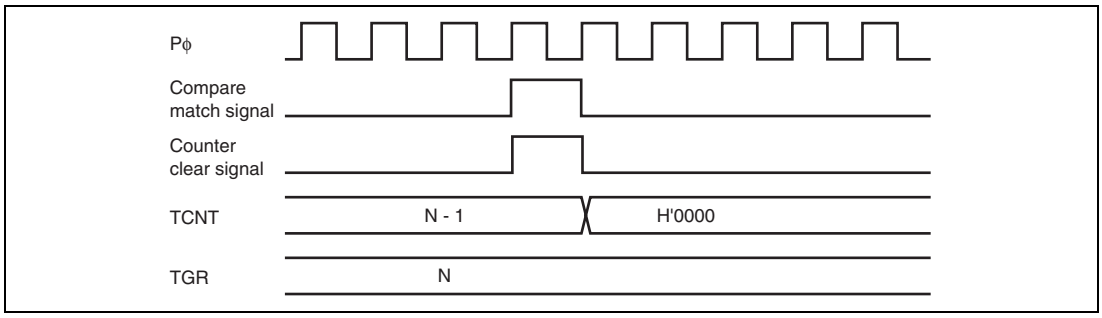


Figure 12.91 Counter Clear Timing (Compare Match) (Channel 5)

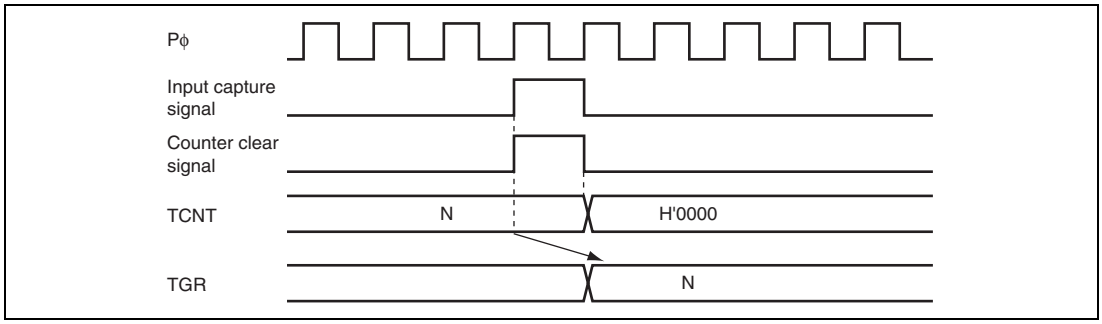


Figure 12.92 Counter Clear Timing (Input Capture) (Channel 0 to Channel 5)

(5) Buffer Operation Timing

Figures 12.93 to 12.95 show the timing in buffer operation.

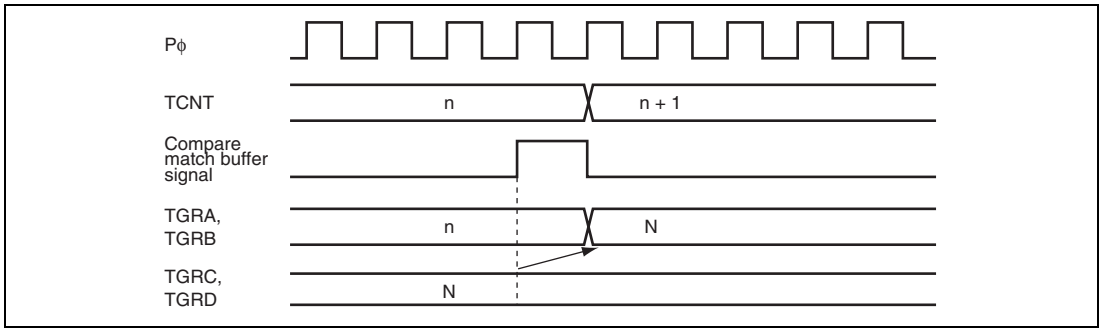


Figure 12.93 Buffer Operation Timing (Compare Match)

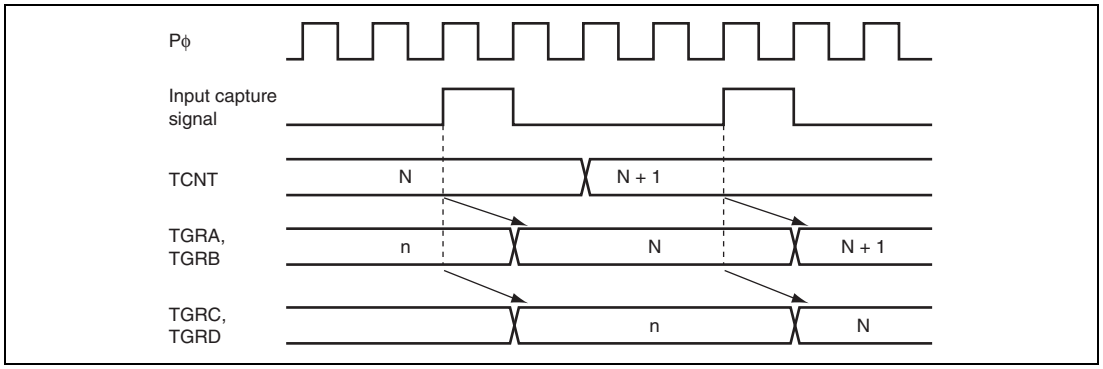


Figure 12.94 Buffer Operation Timing (Input Capture)

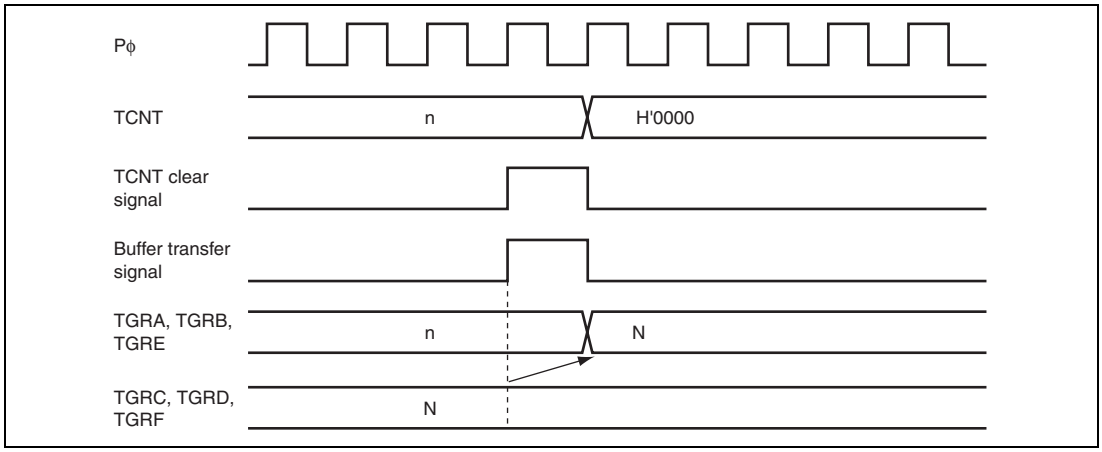


Figure 12.95 Buffer Transfer Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figures 12.96 to 12.98 show the buffer transfer timing in complementary PWM mode.

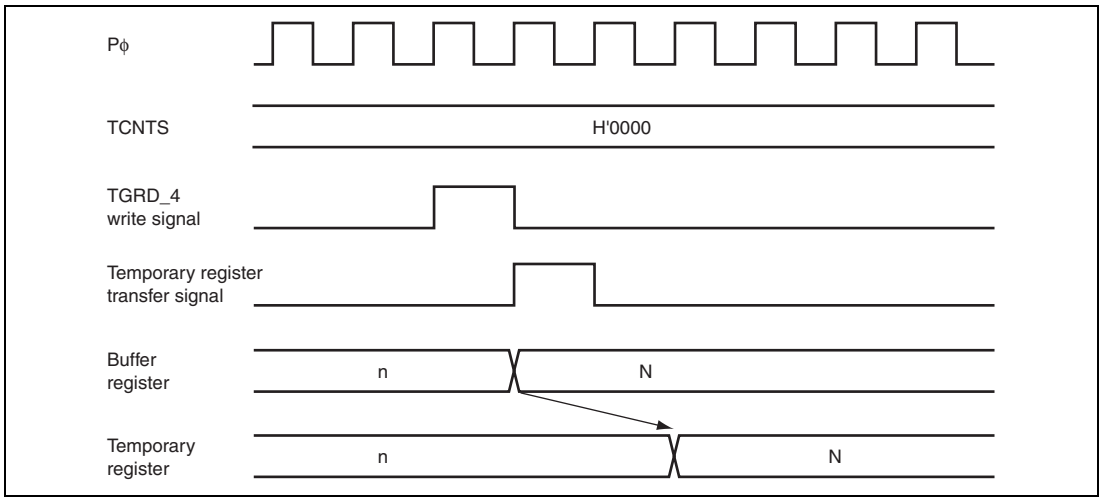


Figure 12.96 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

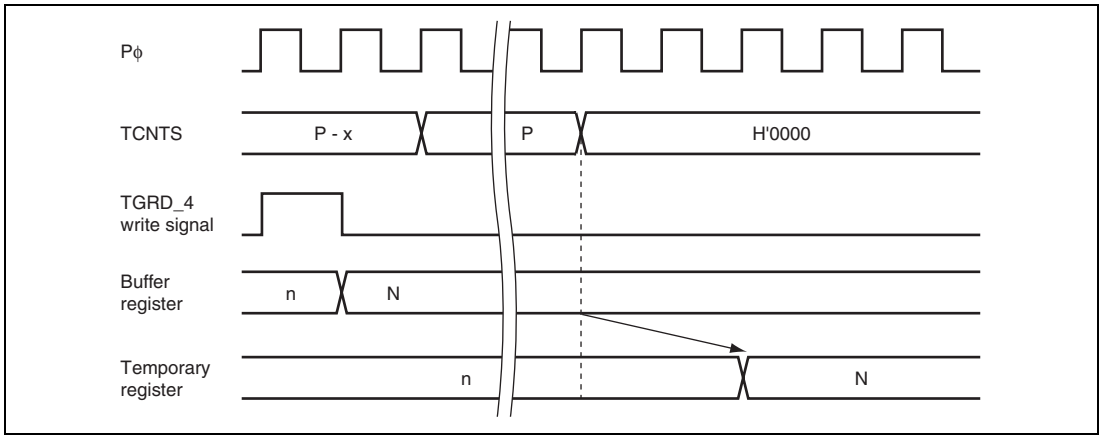


Figure 12.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

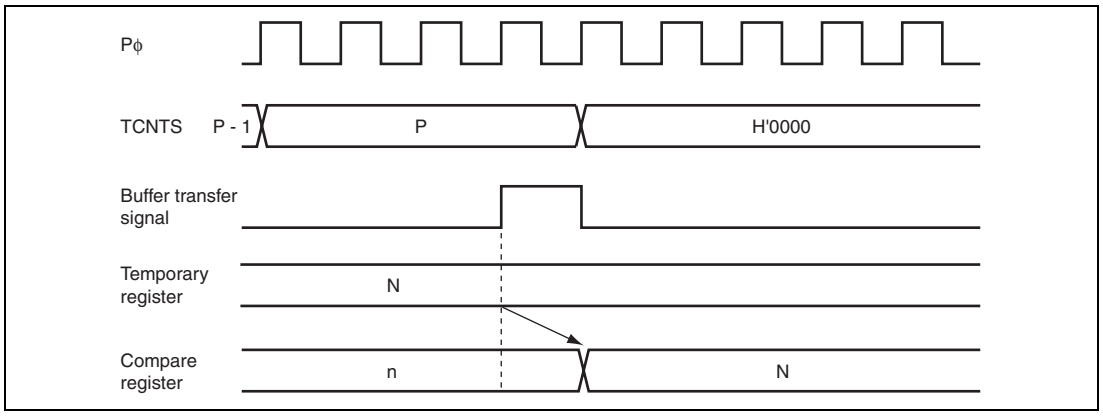


Figure 12.98 Transfer Timing from Temporary Register to Compare Register

12.6.2 Interrupt Signal Timing

(1) TGF Flag Setting Timing in Case of Compare Match

Figures 12.99 and 12.100 show the timing for setting of the TGF flag in TSR on compare match, and TGI interrupt request signal timing.

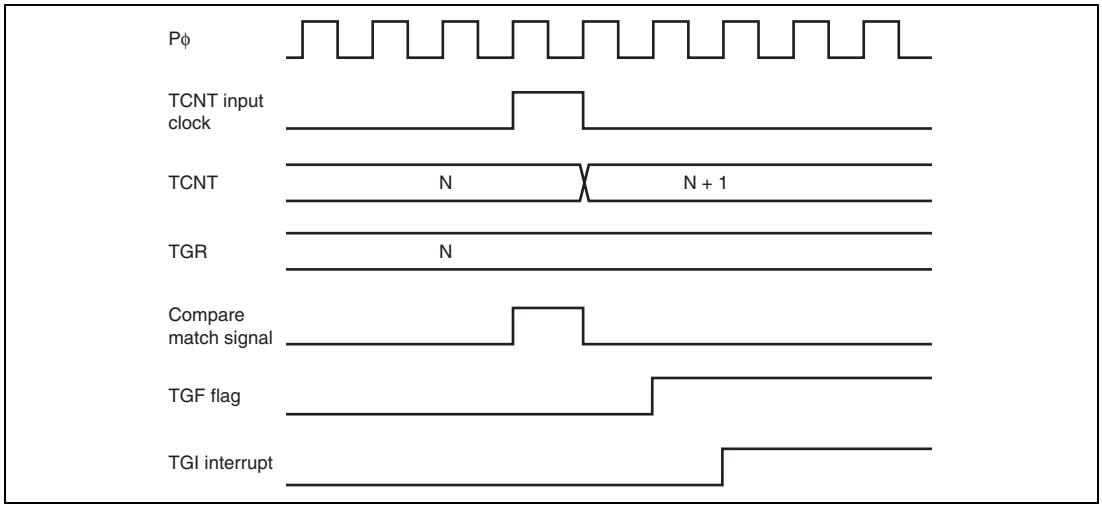


Figure 12.99 TGI Interrupt Timing (Compare Match) (Channels 0 to 4)

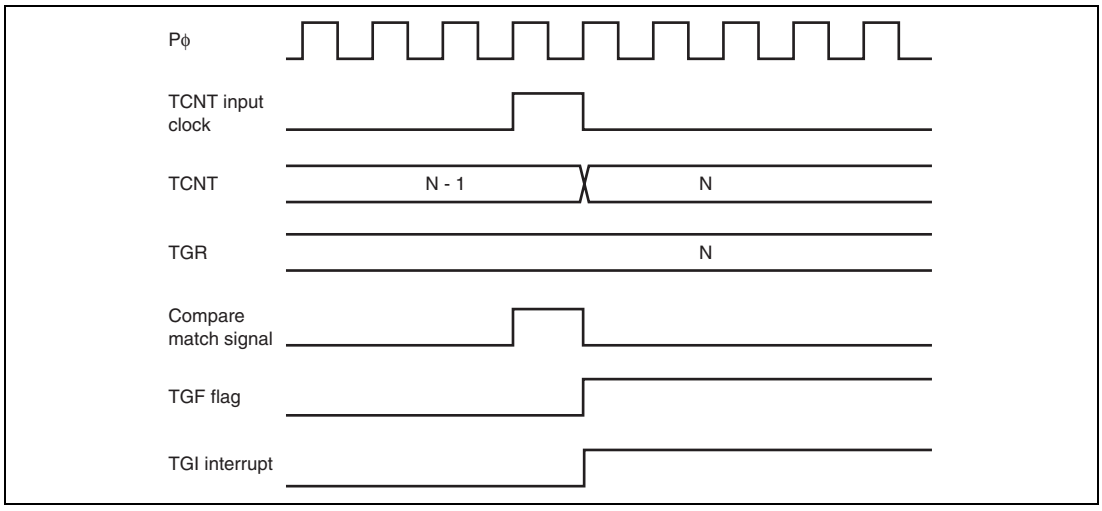


Figure 12.100 TGI Interrupt Timing (Compare Match) (Channel 5)

(2) TGF Flag Setting Timing in Case of Input Capture

Figures 12.101 and 12.102 show the timing for setting of the TGF flag in TSR on input capture, and TGI interrupt request signal timing.

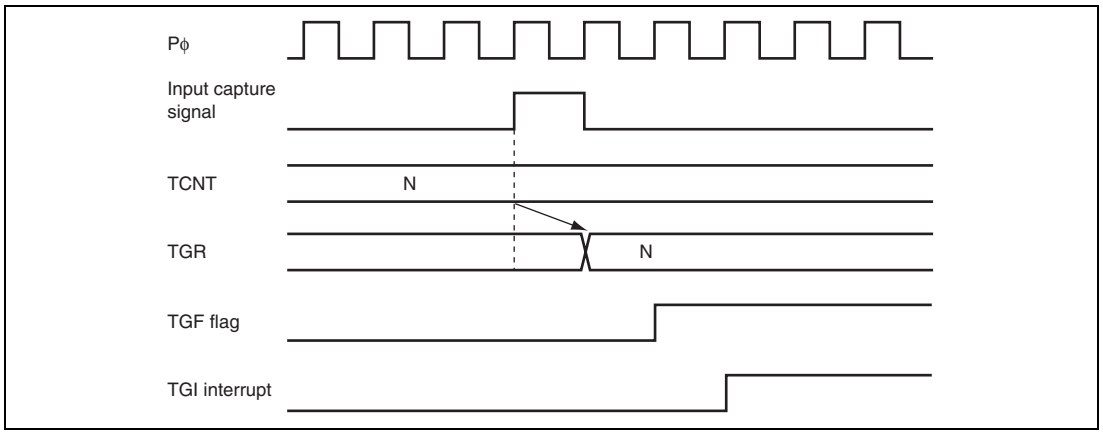


Figure 12.101 TGI Interrupt Timing (Input Capture) (Channels 0 to 4)

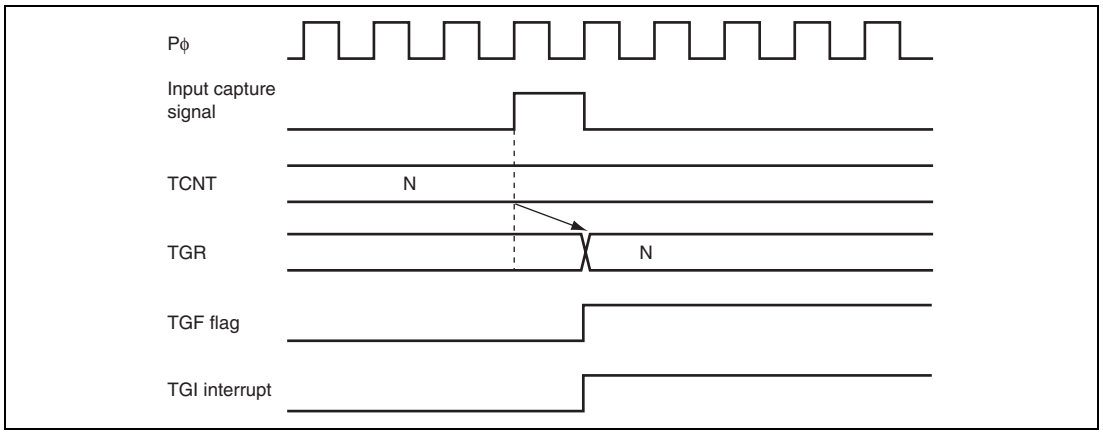


Figure 12.102 TGI Interrupt Timing (Input Capture) (Channel 5)

(3) TCFV Flag/TCFU Flag Setting Timing

Figure 12.103 shows the timing for setting of the TCFV flag in TSR on overflow, and TCIV interrupt request signal timing.

Figure 12.104 shows the timing for setting of the TCFU flag in TSR on underflow, and TCIU interrupt request signal timing.

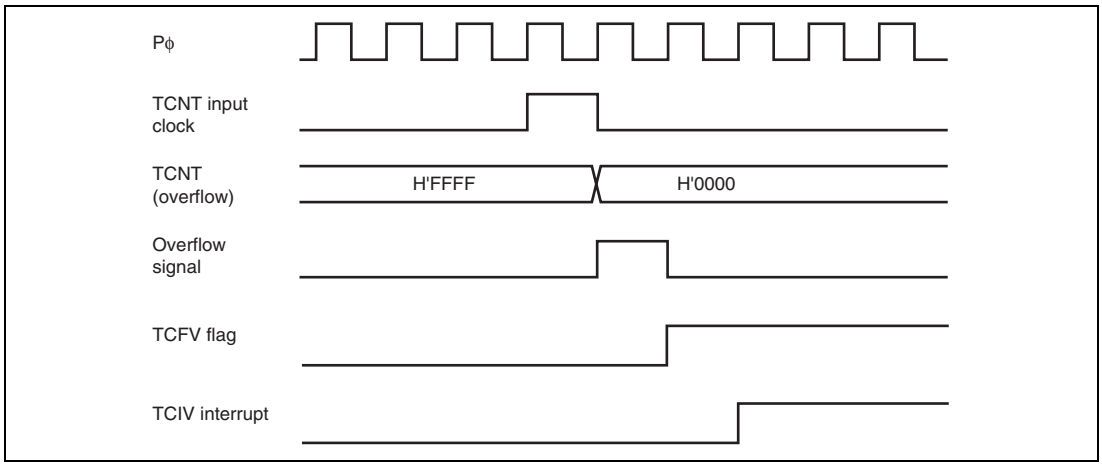


Figure 12.103 TCIV Interrupt Setting Timing

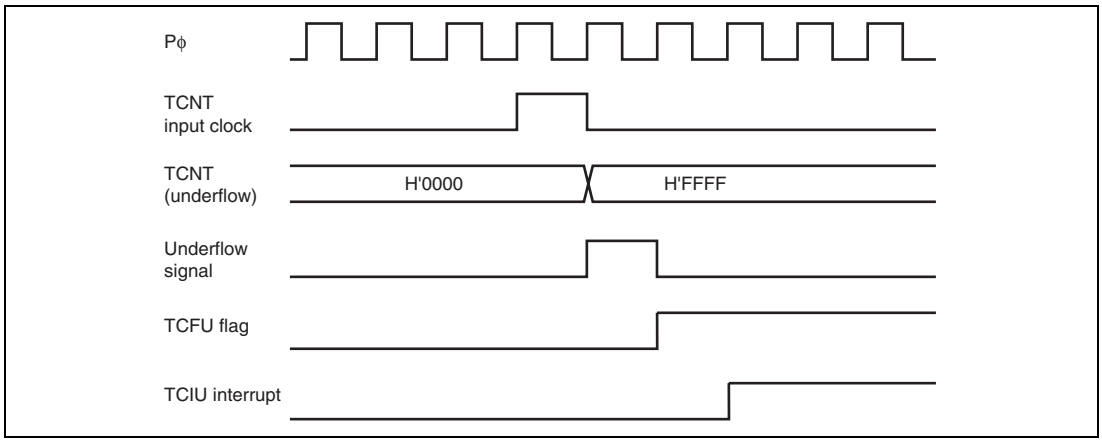


Figure 12.104 TCIU Interrupt Setting Timing

(4) Status Flag Clearing Timing

After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. Figures 12.105 and 12.106 show the timing for status flag clearing by the CPU, and figure 12.107 show the timing for status flag clearing by the DMAC.

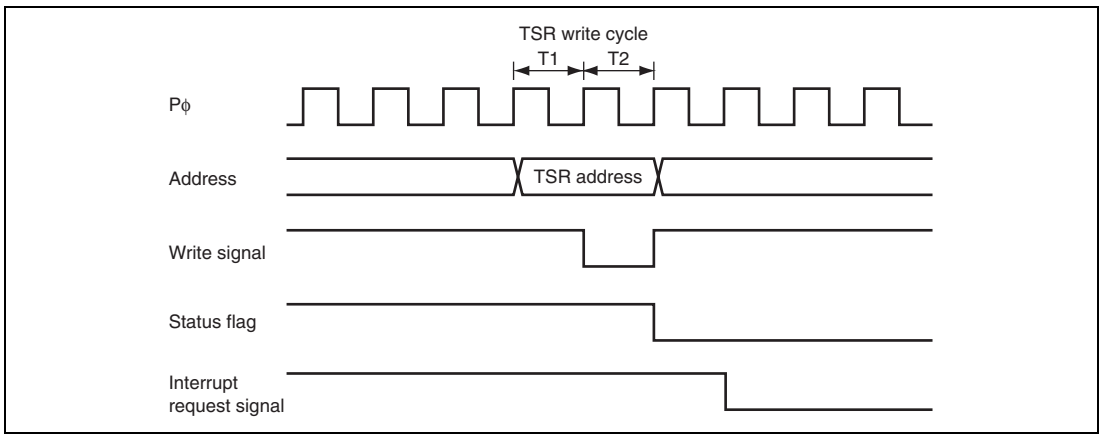


Figure 12.105 Timing for Status Flag Clearing by CPU (Channels 0 to 4)

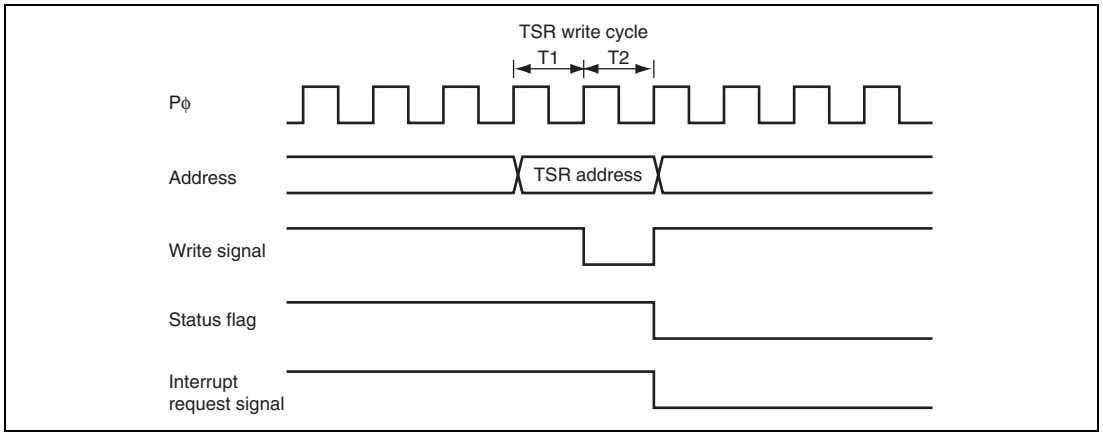


Figure 12.106 Timing for Status Flag Clearing by CPU (Channel 5)

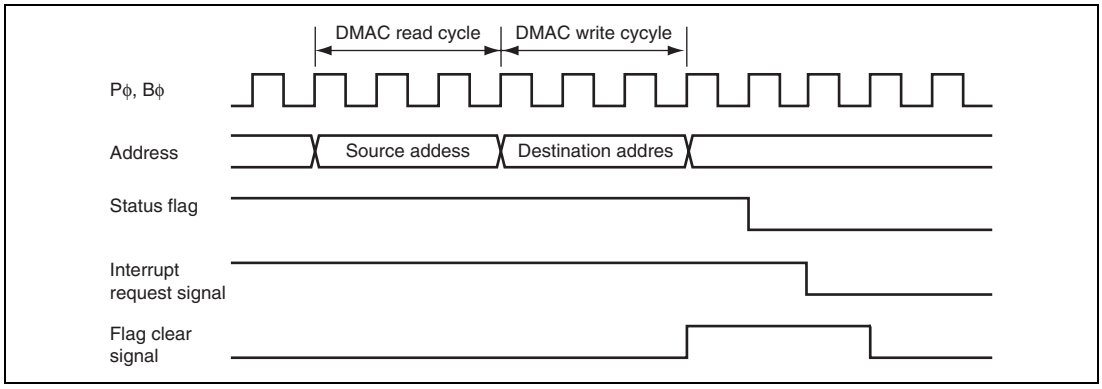


Figure 12.107 Timing for Status Flag Clearing by DMAC Activation (Channels 0 to 4)

12.7 Usage Notes

12.7.1 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 25, Power-Down Modes.

12.7.2 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 12.108 shows the input clock conditions in phase counting mode.

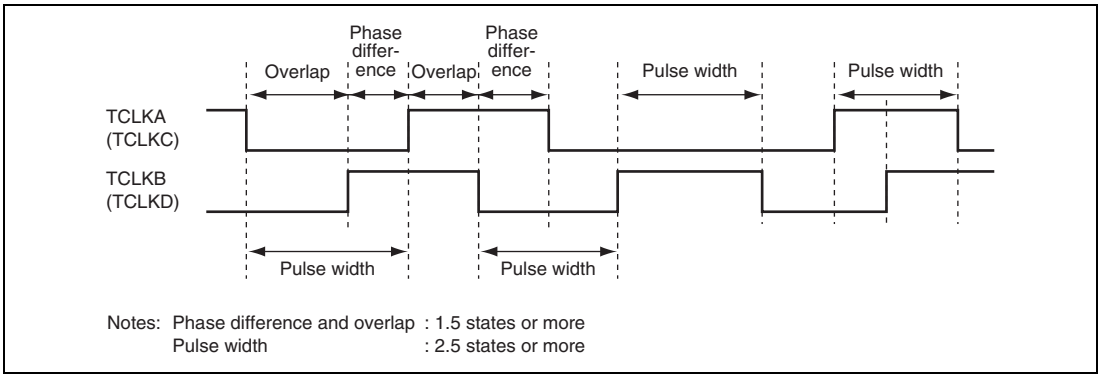


Figure 12.108 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

12.7.3 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:

- Channels 0 to 4

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

- Channel 5

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

Where f: Counter frequency
Pϕ: MTU2 peripheral clock operating frequency
N: TGR set value

12.7.4 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 12.109 shows the timing in this case.

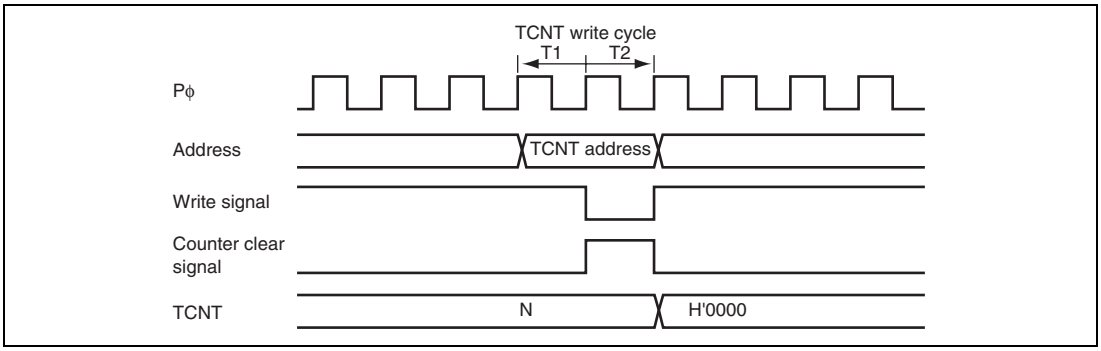


Figure 12.109 Contention between TCNT Write and Clear Operations

12.7.5 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 12.110 shows the timing in this case.

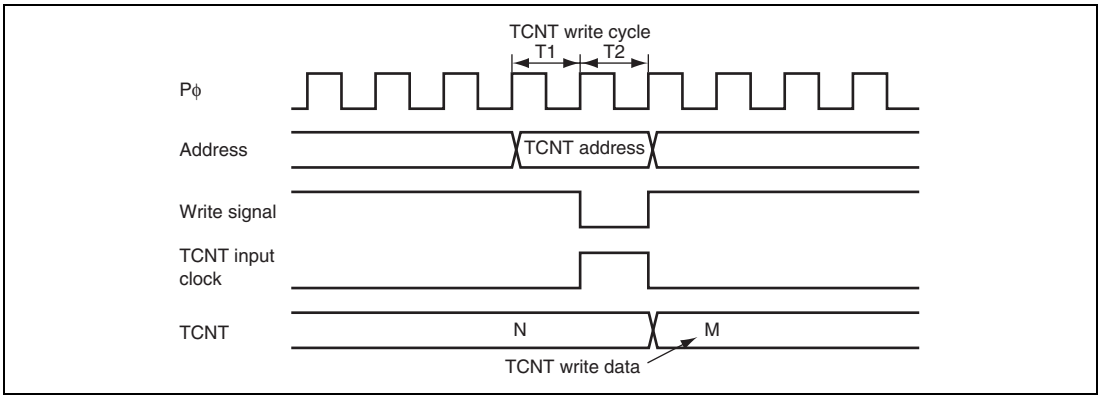


Figure 12.110 Contention between TCNT Write and Increment Operations

12.7.6 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 12.111 shows the timing in this case.

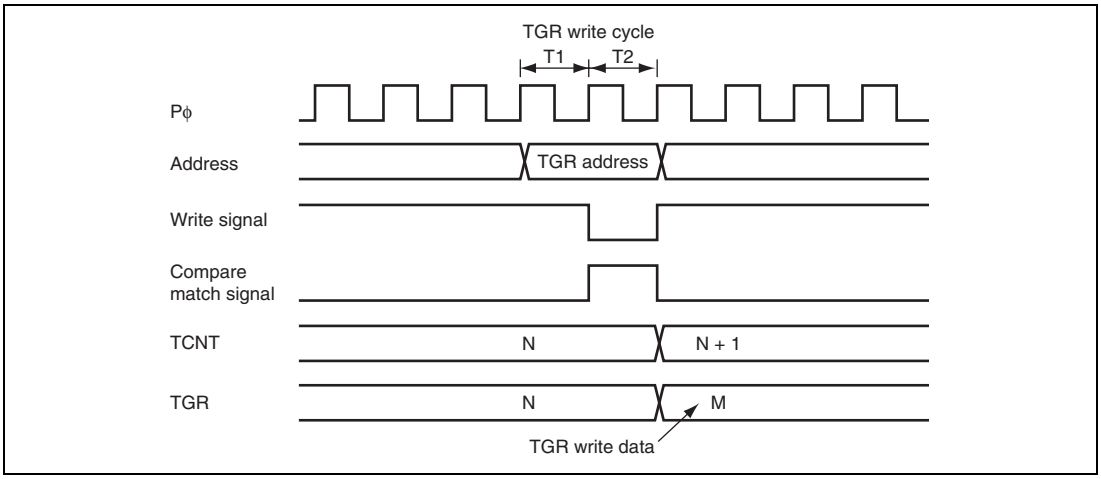


Figure 12.111 Contention between TGR Write and Compare Match

12.7.7 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 before write.

Figure 12.112 shows the timing in this case.

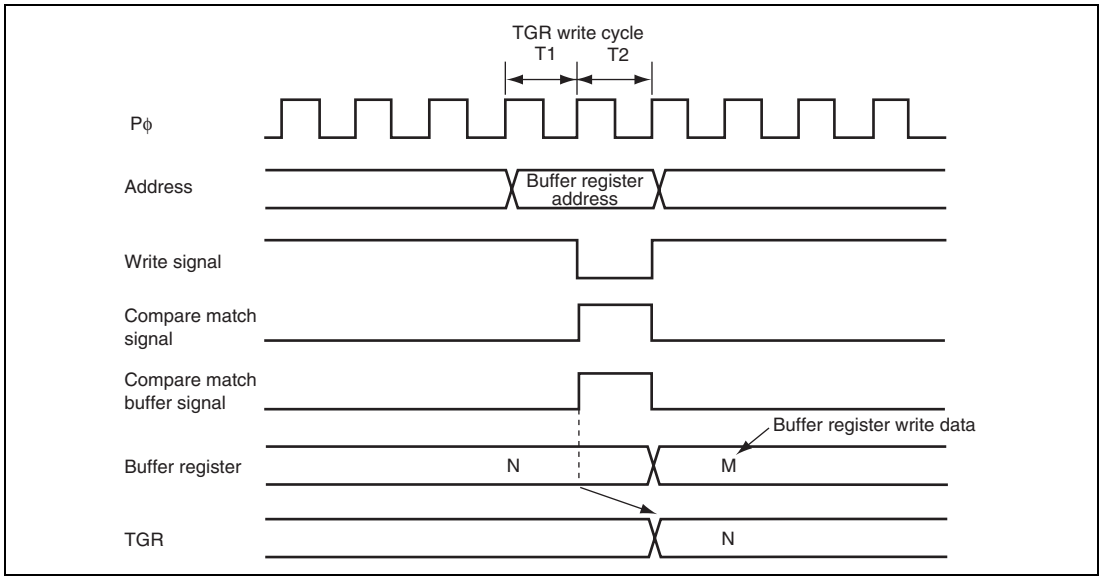


Figure 12.112 Contention between Buffer Register Write and Compare Match

12.7.8 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 12.113 shows the timing in this case.

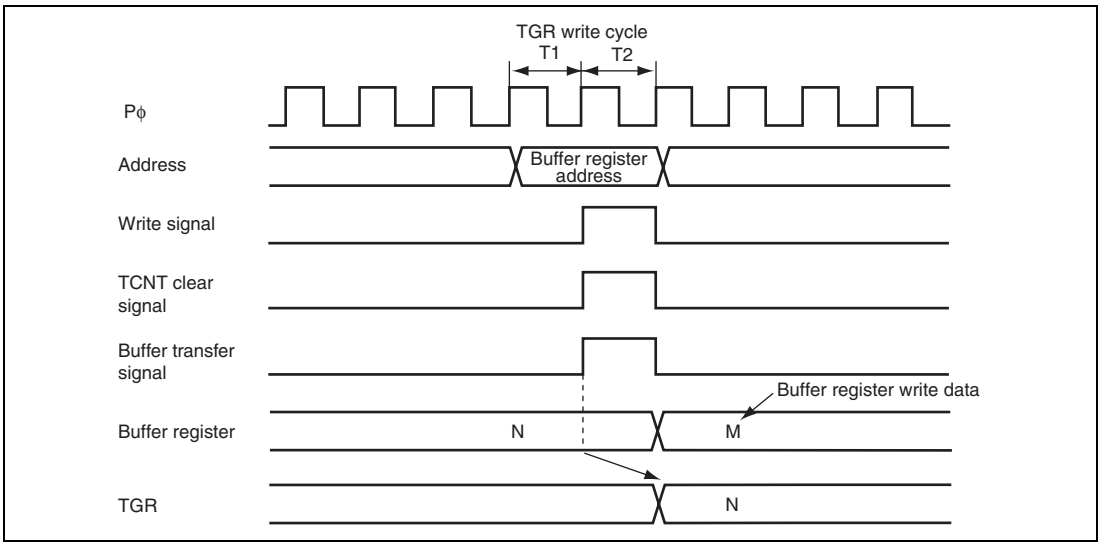


Figure 12.113 Contention between Buffer Register Write and TCNT Clear

12.7.9 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 12.114 and 12.115 show the timing in this case.

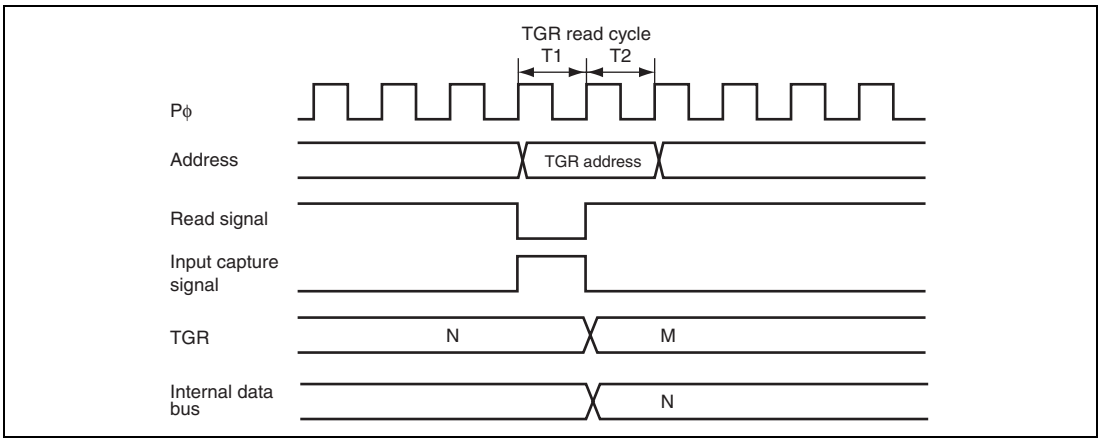


Figure 12.114 Contention between TGR Read and Input Capture (Channels 0 to 4)

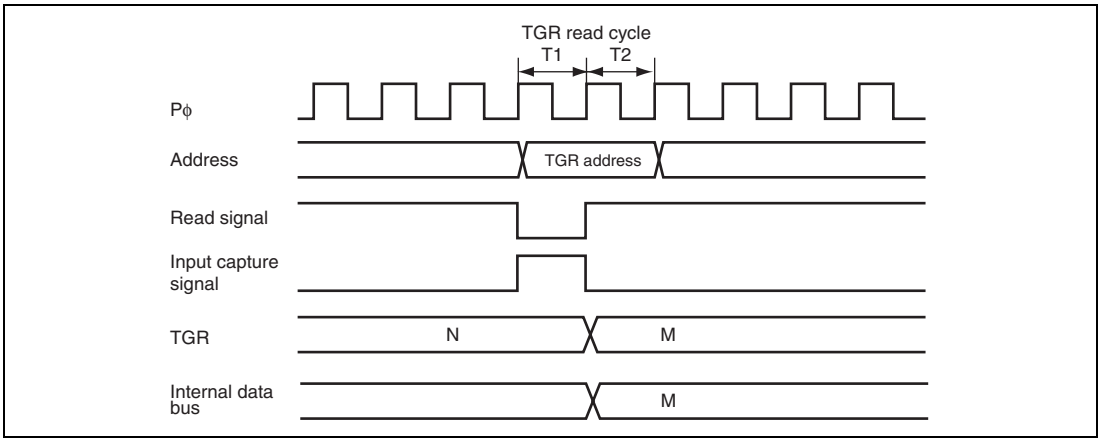


Figure 12.115 Contention between TGR Read and Input Capture (Channel 5)

12.7.10 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 12.116 and 12.117 show the timing in this case.

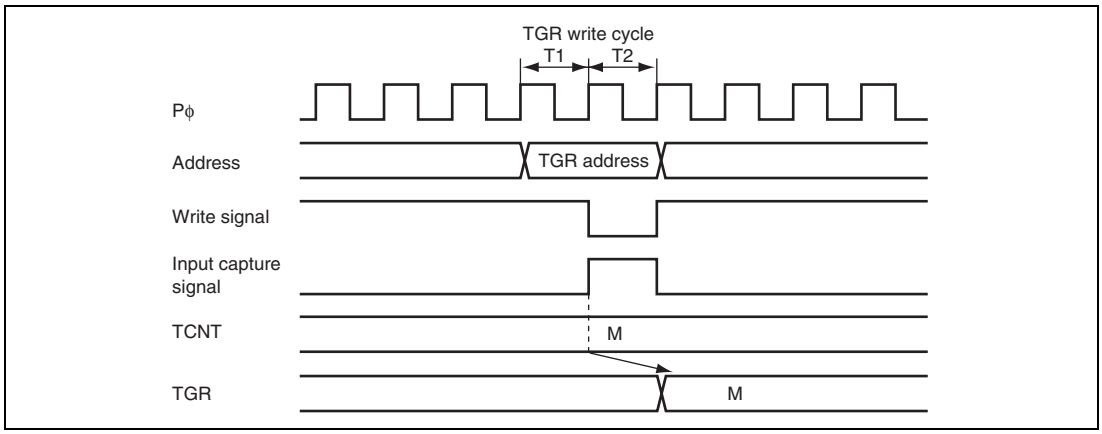


Figure 12.116 Contention between TGR Write and Input Capture (Channels 0 to 4)

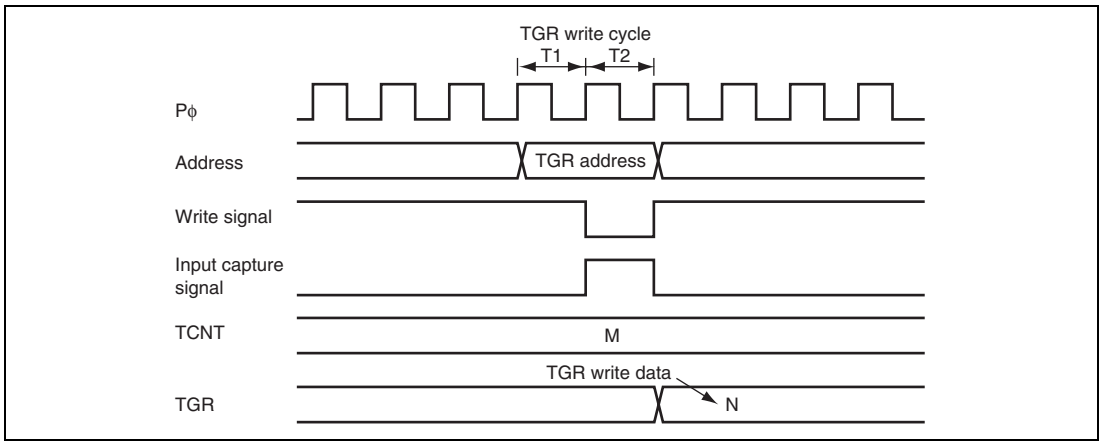


Figure 12.117 Contention between TGR Write and Input Capture (Channel 5)

12.7.11 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 12.118 shows the timing in this case.

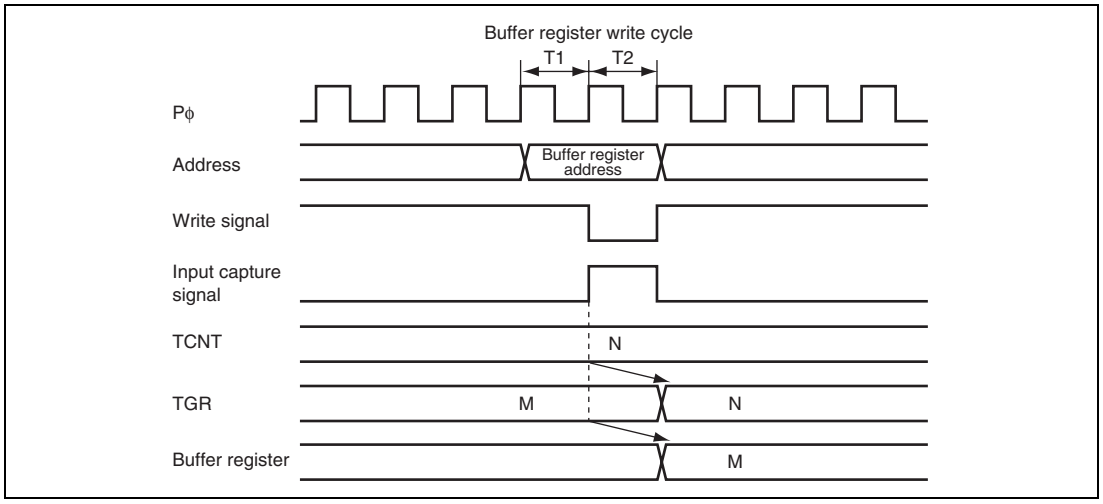


Figure 12.118 Contention between Buffer Register Write and Input Capture

12.7.12 TCNT_2 Write and Overflow/Underflow Contention in Cascade Connection

With timer counters TCNT_1 and TCNT_2 in a cascade connection, when a contention occurs during TCNT_1 count (during a TCNT_2 overflow/underflow) in the T2 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 TGRD_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 12.119.

For cascade connections, be sure to synchronize settings for channels 1 and 2 when setting TCNT clearing.

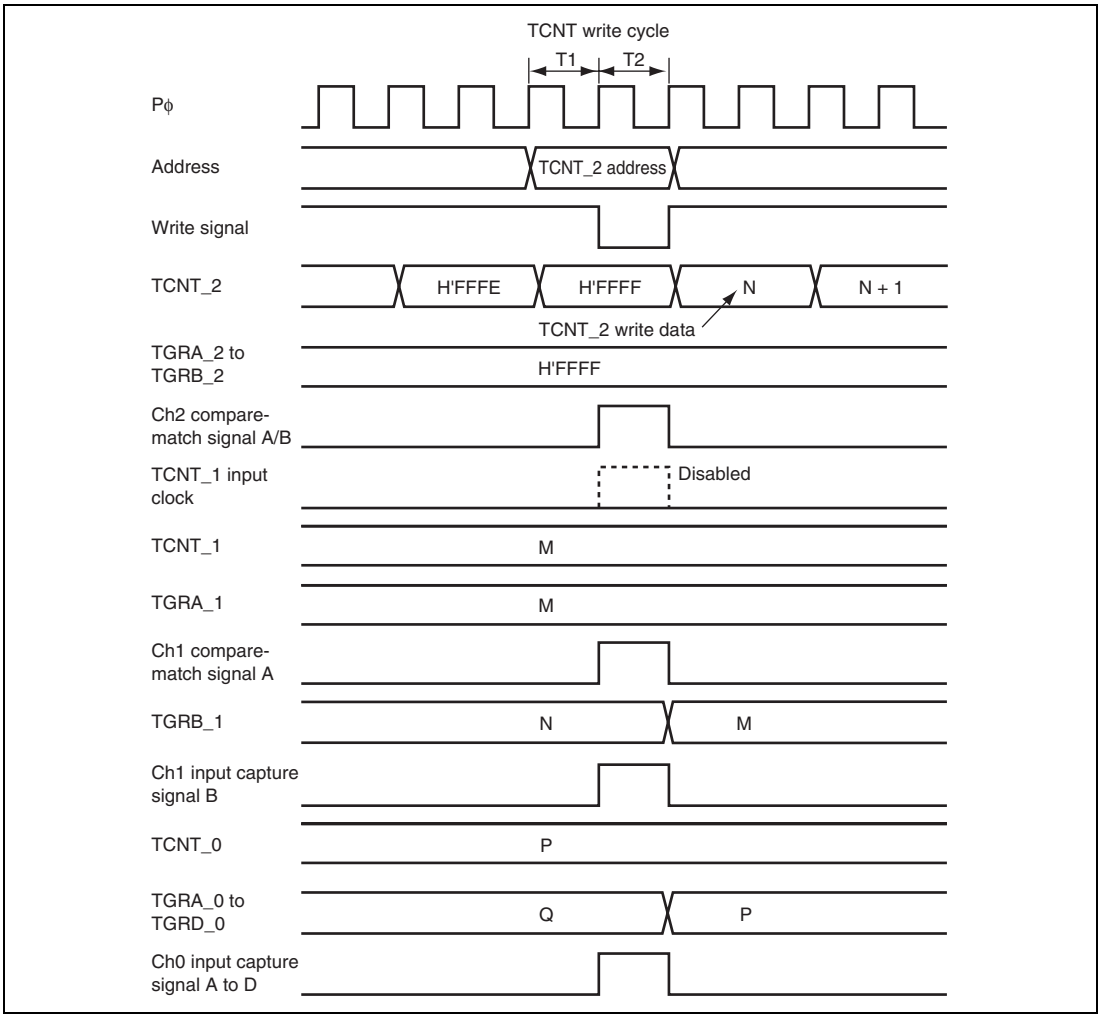


Figure 12.119 TCNT_2 Write and Overflow/Underflow Contention with Cascade Connection

12.7.13 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 12.120.

When counting begins in another operating mode, be sure that TCNT_3 and TCNT_4 are set to the initial values.

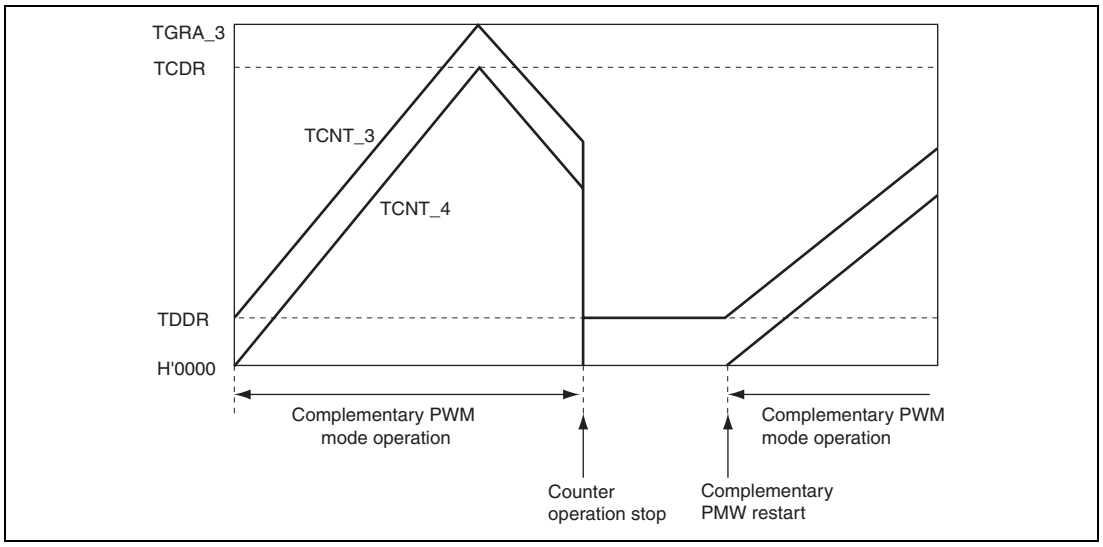


Figure 12.120 Counter Value during Complementary PWM Mode Stop

12.7.14 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 the BFA bit in TMDR_3 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.

12.7.15 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 in TMDR_4 to 0. The TIOC4C pin will be unable to produce its waveform output if the BFA bit in 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 in 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 in TSR_3 and TSR_4 are not set when TGRC_3 and TGRD_3 are operating as buffer registers.

Figure 12.121 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.

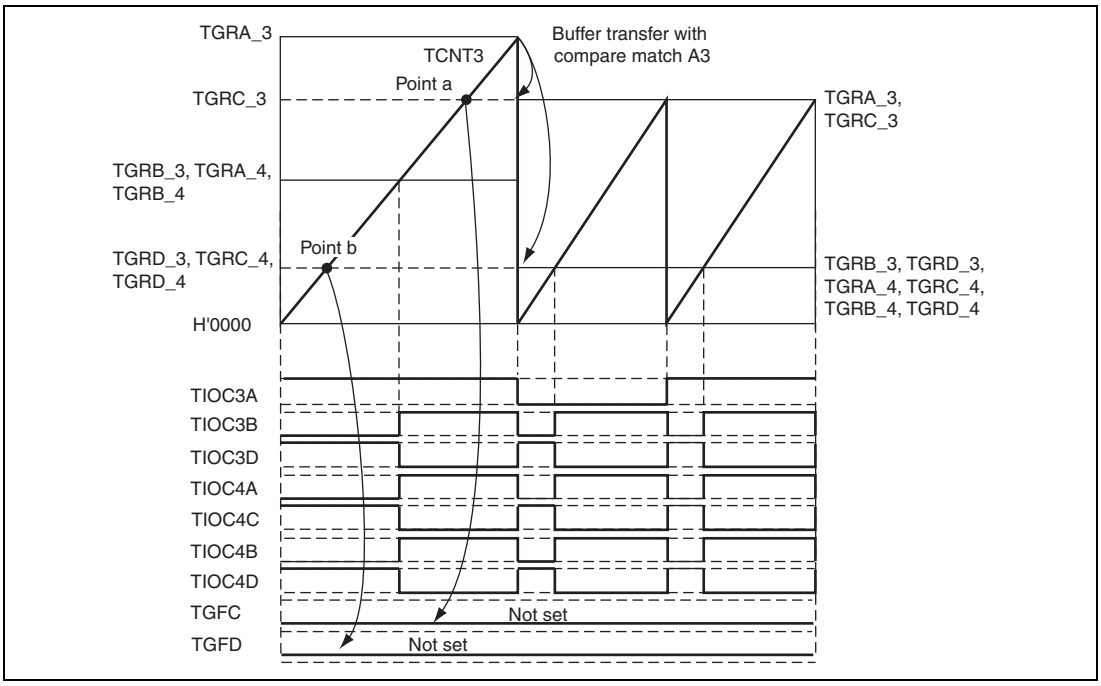


Figure 12.121 Buffer Operation and Compare-Match Flags in Reset Synchronous PWM Mode

12.7.16 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 in 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 12.122 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.

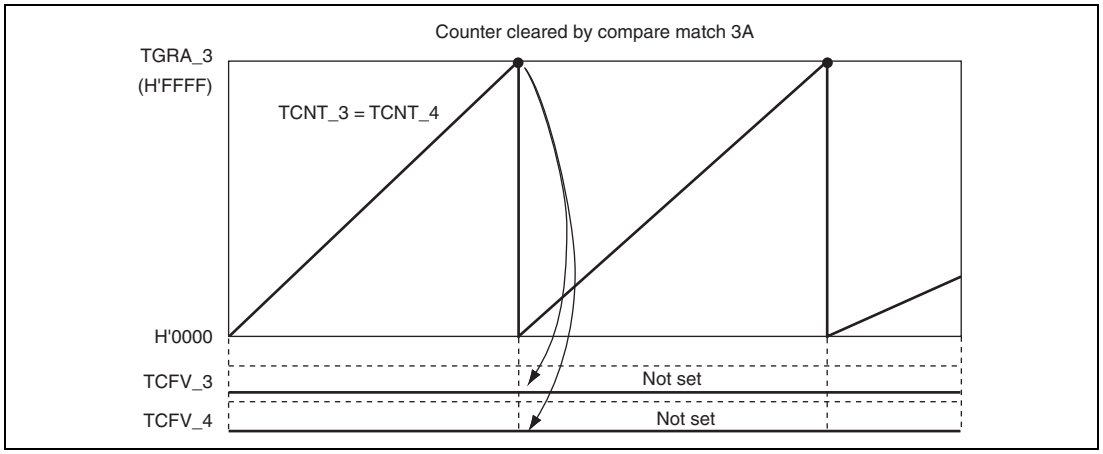


Figure 12.122 Reset Synchronous PWM Mode Overflow Flag

12.7.17 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 12.123 shows the operation timing when a TGR compare match is specified as the clearing source, and when H'FFFF is set in TGR.

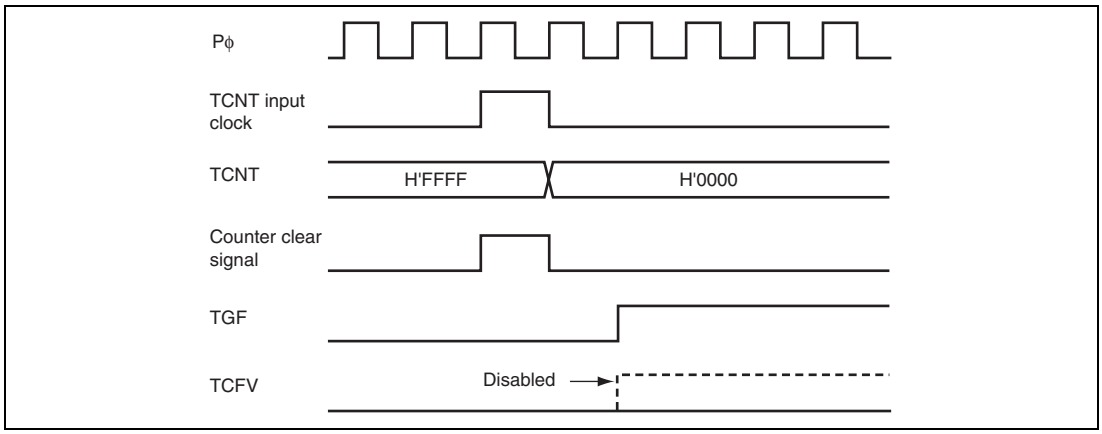


Figure 12.123 Contention between Overflow and Counter Clearing

12.7.18 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 12.124 shows the operation timing when there is contention between TCNT write and overflow.

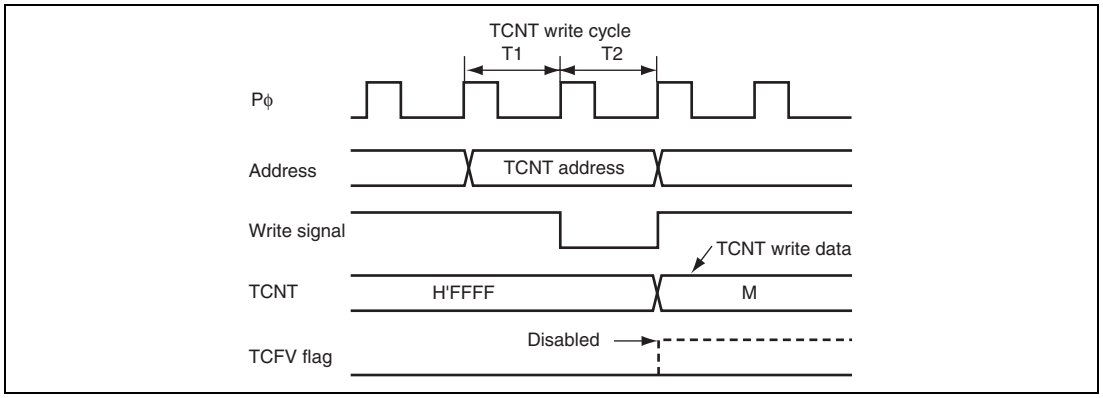


Figure 12.124 Contention between TCNT Write and Overflow

12.7.19 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.

12.7.20 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.

12.7.21 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.

12.7.22 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.

12.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode

In complementary PWM mode, when output waveform control during synchronous counter clearing is enabled (WRE in the TWCR register set to 1), the following problems may occur when condition (1) or condition (2), below, is satisfied.

- Dead time for the PWM output pins may be too short (or nonexistent).
- Active-level output from the PWM negative-phase pins may occur outside the correct active-level output interval

Condition (1): When synchronous clearing occurs in the PWM output dead time interval within initial output suppression interval (10) (figure 12.125).

Condition (2): When synchronous clearing occurs within initial output suppression interval (10) or (11) and TGRB_3 ≤ TDDR, TGRA_4 ≤ TDDR, or TGRB_4 ≤ TDDR is true (figure 12.126)

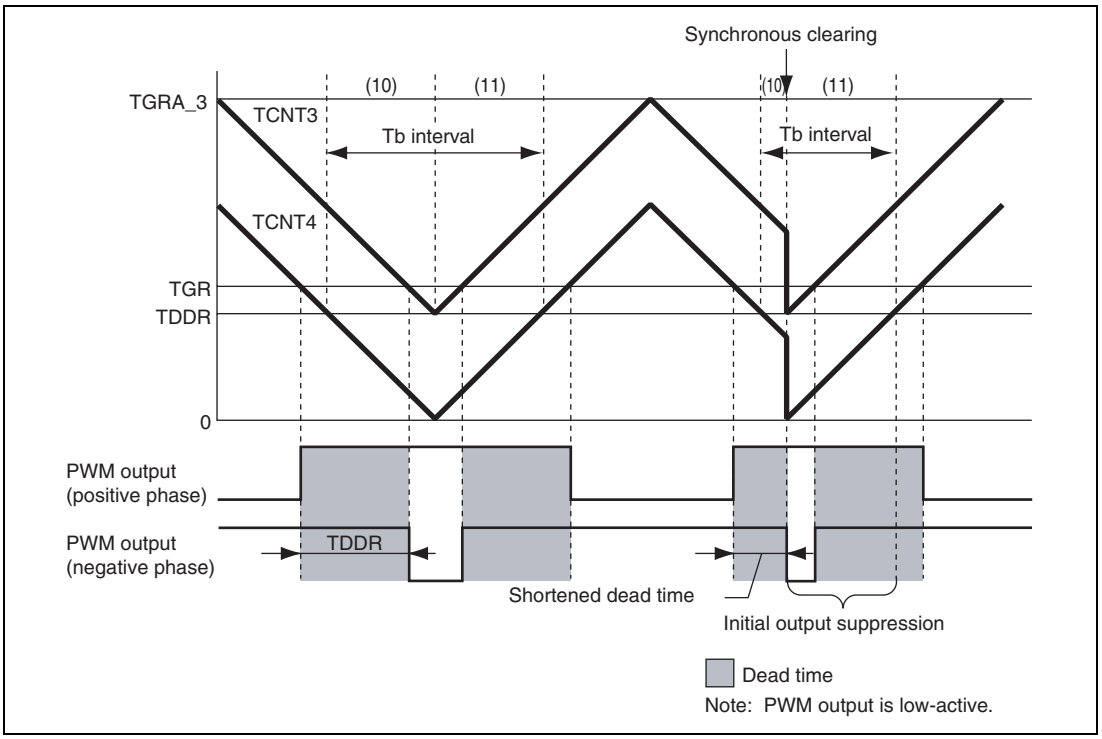


Figure 12.125 Condition (1) Synchronous Clearing Example

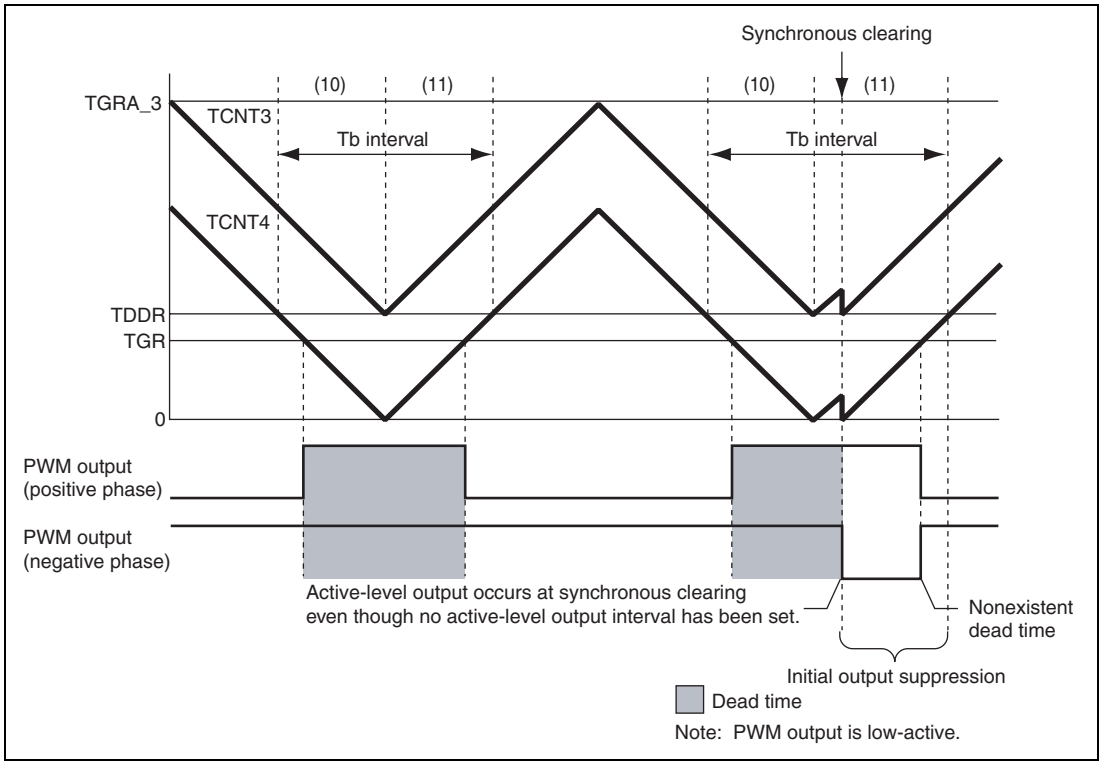


Figure 12.126 Condition (2) Synchronous Clearing Example

The following workaround can be used to avoid these problems.

When using synchronous clearing, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR.

12.8 MTU2 Output Pin Initialization

12.8.1 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.

12.8.2 Reset Start Operation

The MTU2 output pins (TIOC*) are initialized low by a power-on reset. 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 power-on 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 *.

12.8.3 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. 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 12.59.

Table 12.59 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

12.8.4 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation, etc.

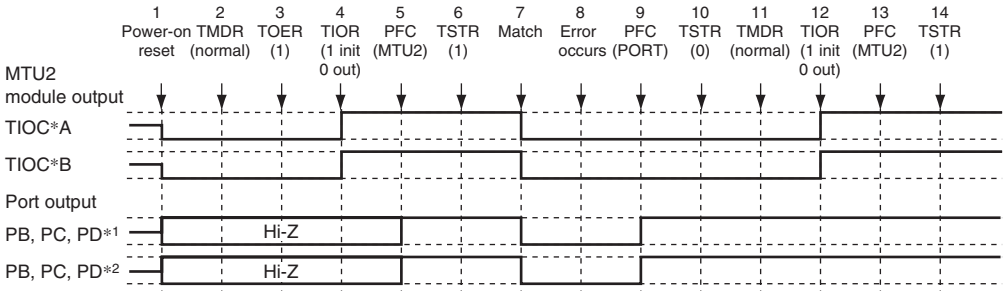
- When making a transition to a mode (Normal, PWM1, PWM2, PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of a TIOR setting.
- In PWM mode 1, since a waveform is not output to the TIOC*B (TIOC *D) pin, setting TIOR will not 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 12.59. 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 12.127 shows an explanatory diagram of the case where an error occurs in normal mode and operation is restarted in normal mode after re-setting.



Notes: 1. This pin is multiplexed with TIOC*A.
2. This pin is multiplexed with TIOC*B.

Figure 12.127 Error Occurrence in Normal Mode, Recovery in Normal Mode

1. After a power-on reset, MTU2 output is low and ports are in the high-impedance state.
2. After a power-on 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 12.128 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.

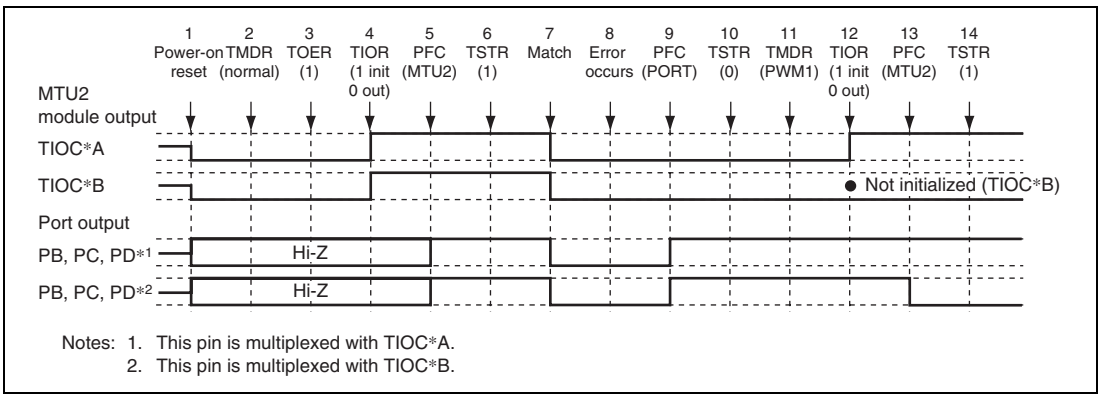


Figure 12.128 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.127.

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 12.129 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.

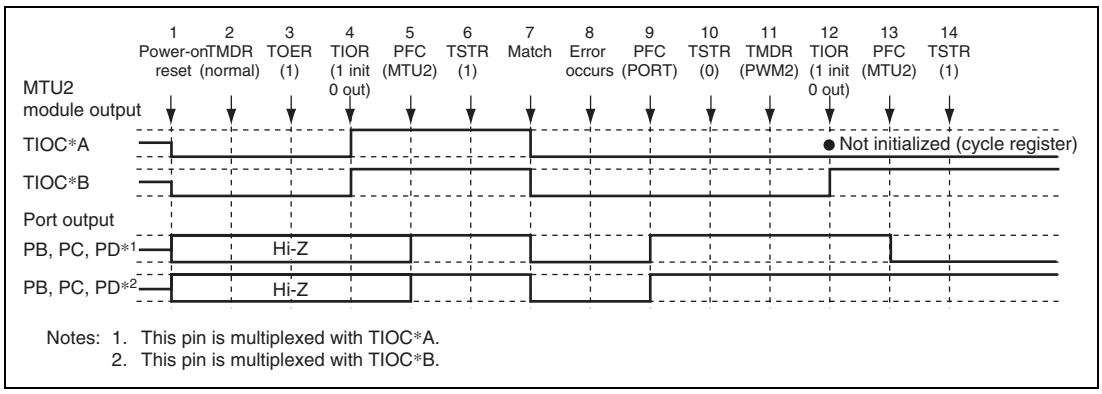


Figure 12.129 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

1 to 10 are the same as in figure 12.127.

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 12.130 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.

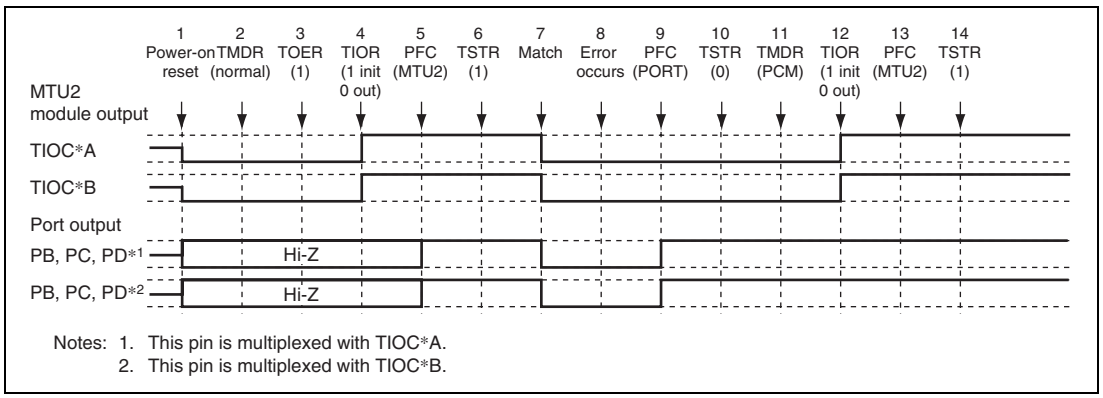


Figure 12.130 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.127.

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 12.131 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.

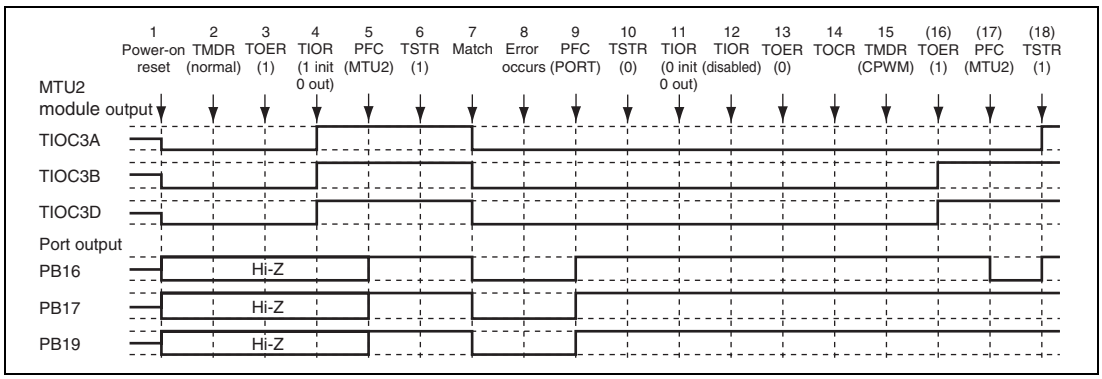


Figure 12.131 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.127.

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 12.132 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.

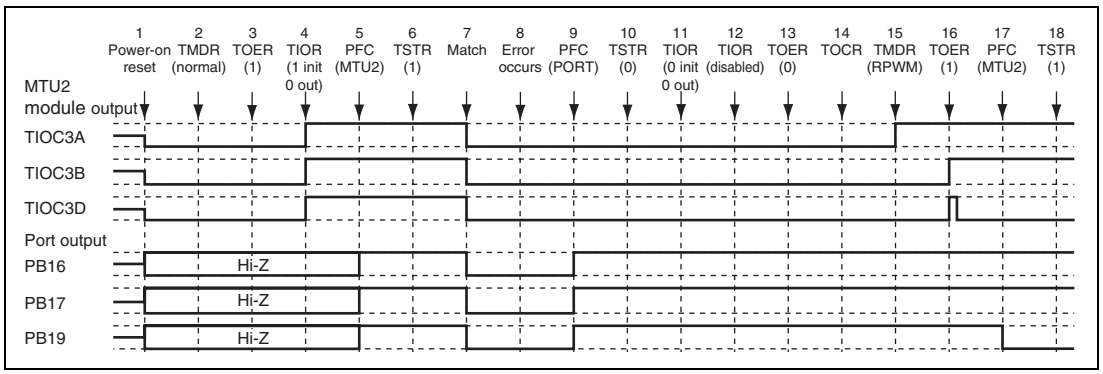


Figure 12.132 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

1 to 13 are the same as in figure 12.127.

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 12.133 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.

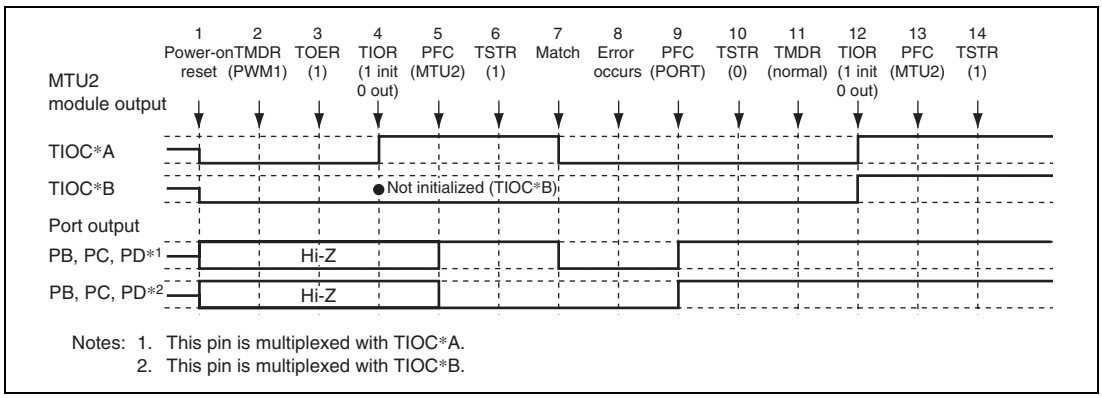


Figure 12.133 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

1. After a power-on 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 12.134 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.

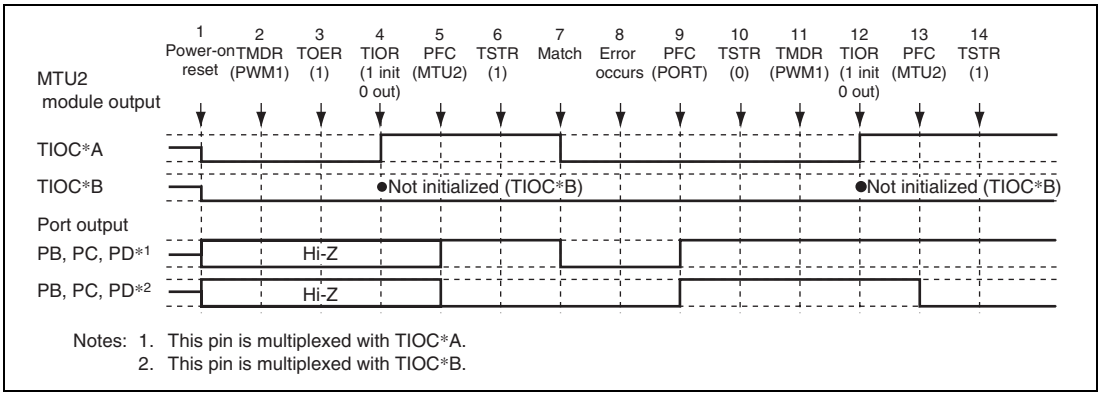


Figure 12.134 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.133.

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 12.135 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.

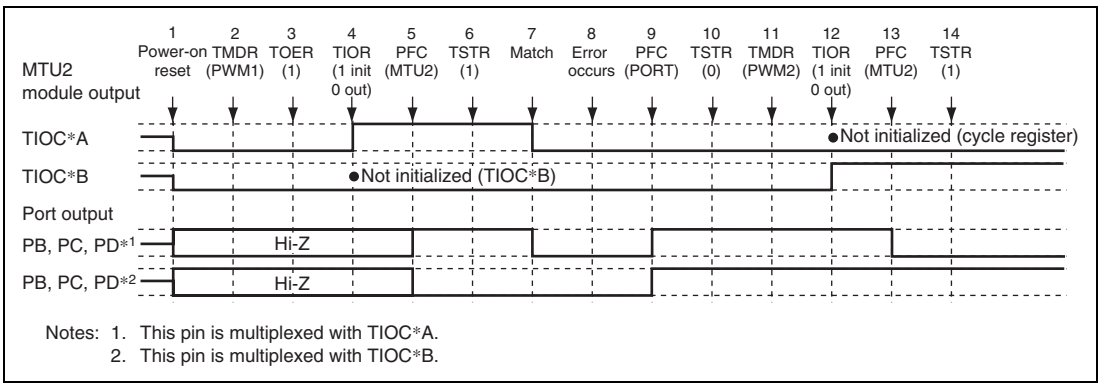


Figure 12.135 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

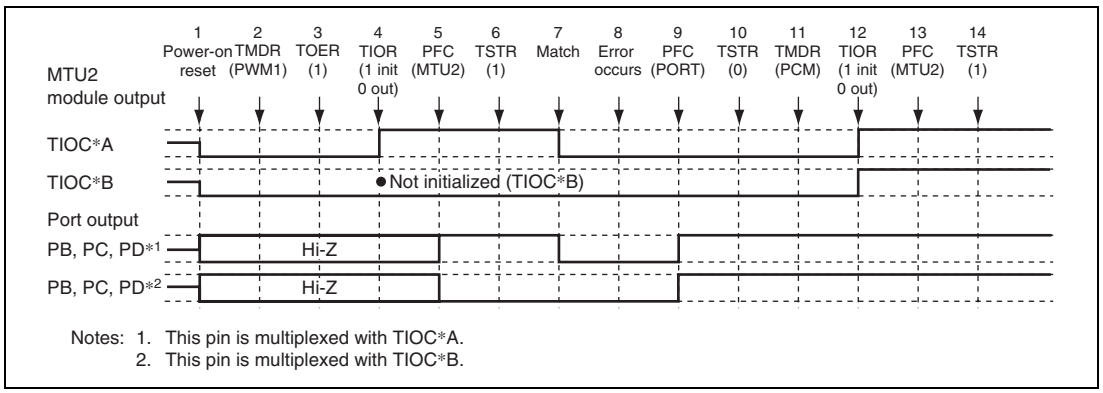
1 to 10 are the same as in figure 12.133.

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 12.136 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.



Notes: 1. This pin is multiplexed with TIOC*A.
2. This pin is multiplexed with TIOC*B.

Figure 12.136 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

1 to 10 are the same as in figure 12.133.

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 12.137 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.

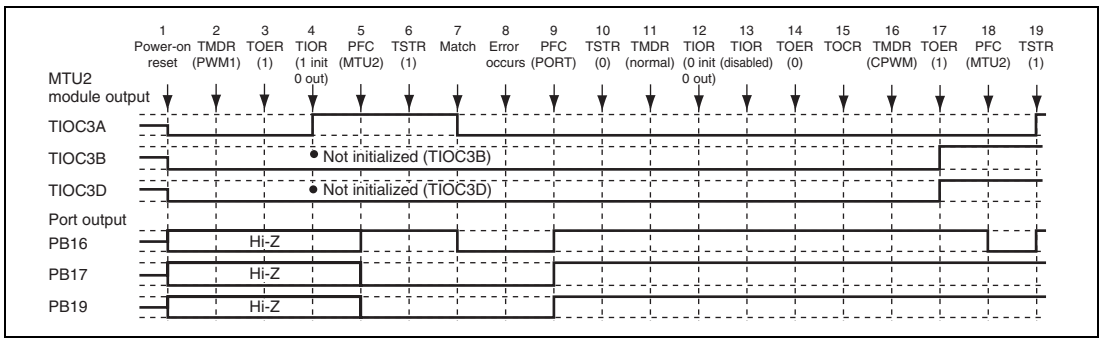


Figure 12.137 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.133.

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 12.138 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.

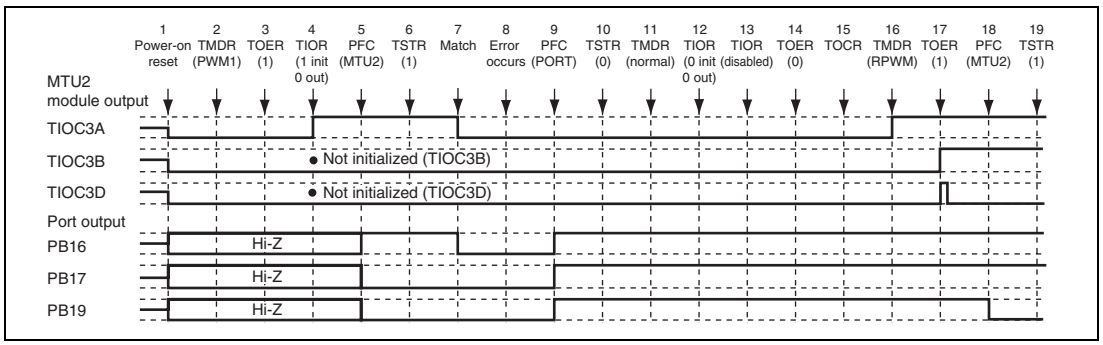


Figure 12.138 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

1 to 14 are the same as in figure 12.137.

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 12.139 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.

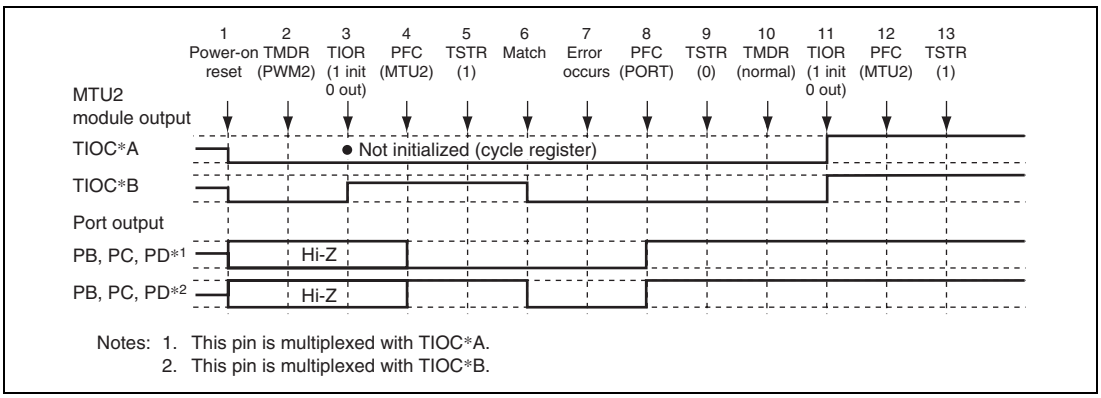


Figure 12.139 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

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 12.140 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.

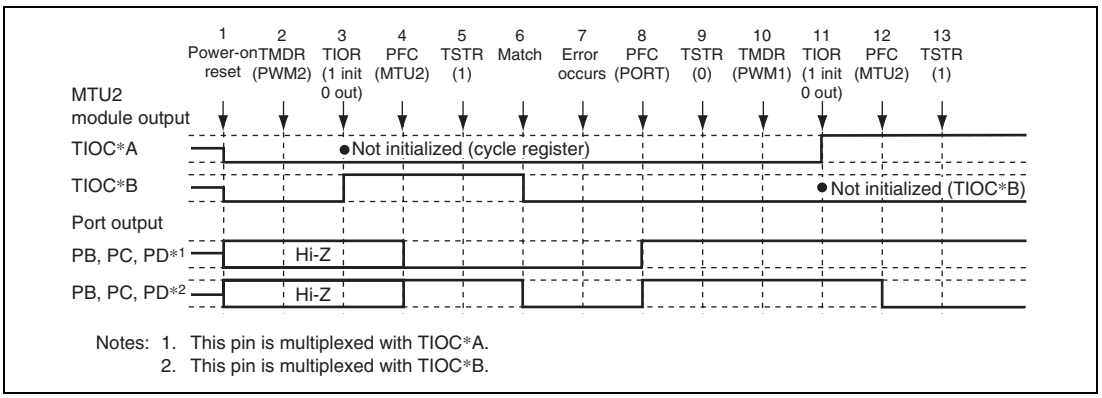


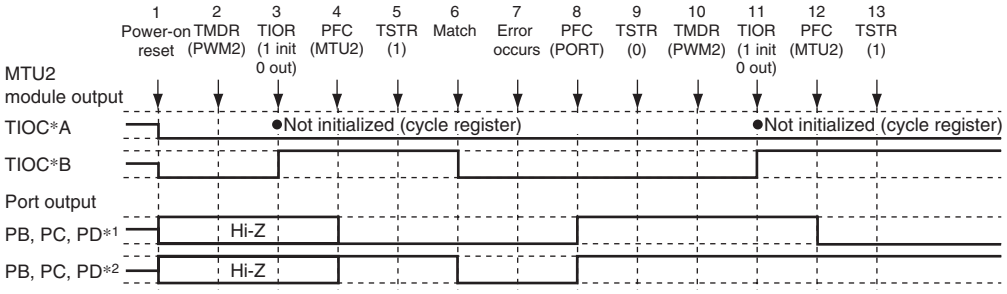
Figure 12.140 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.139.

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 12.141 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.



Notes: 1. This pin is multiplexed with TIOC*A.
2. This pin is multiplexed with TIOC*B.

Figure 12.141 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.139.

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 12.142 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.

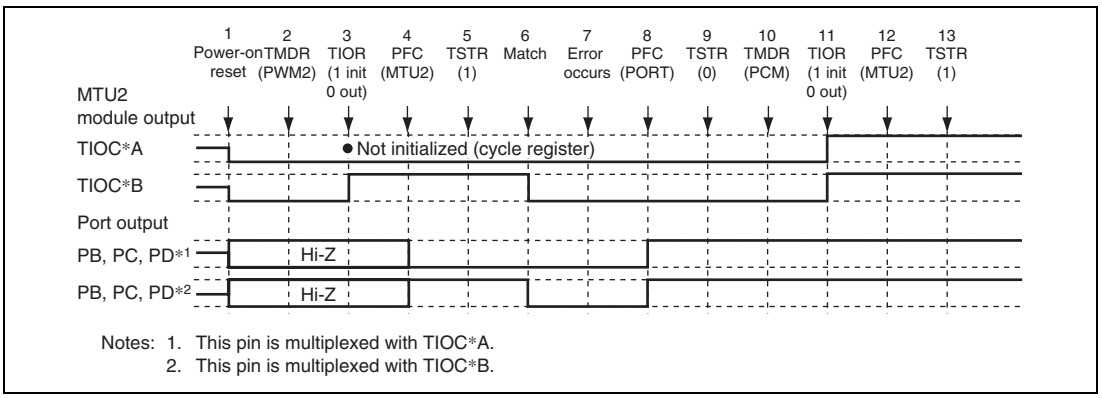


Figure 12.142 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.139.

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 12.143 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.

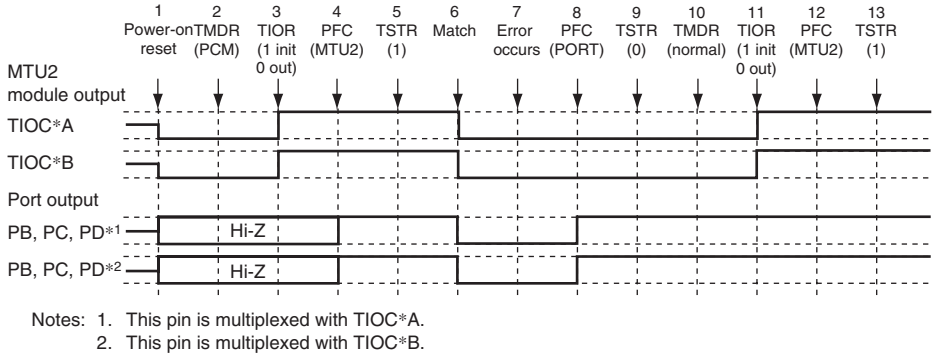


Figure 12.143 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

1. After a power-on 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 12.144 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.

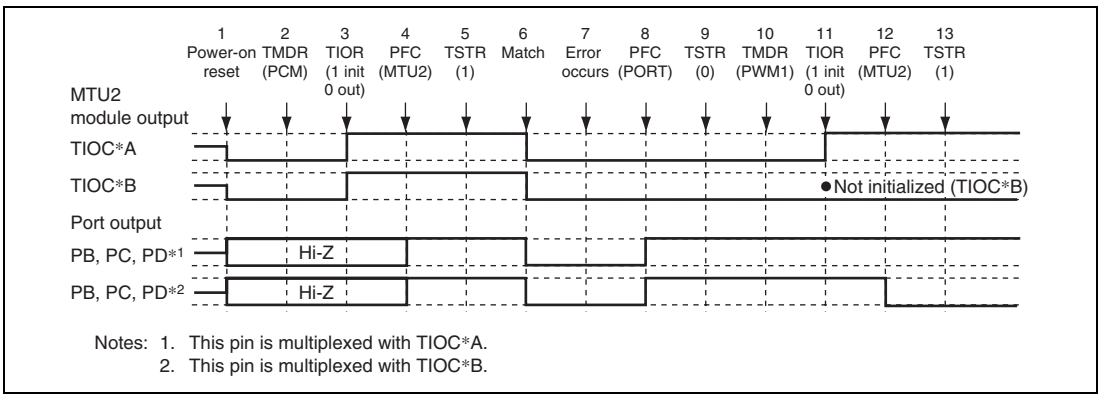


Figure 12.144 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

1 to 9 are the same as in figure 12.143.

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 12.145 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.

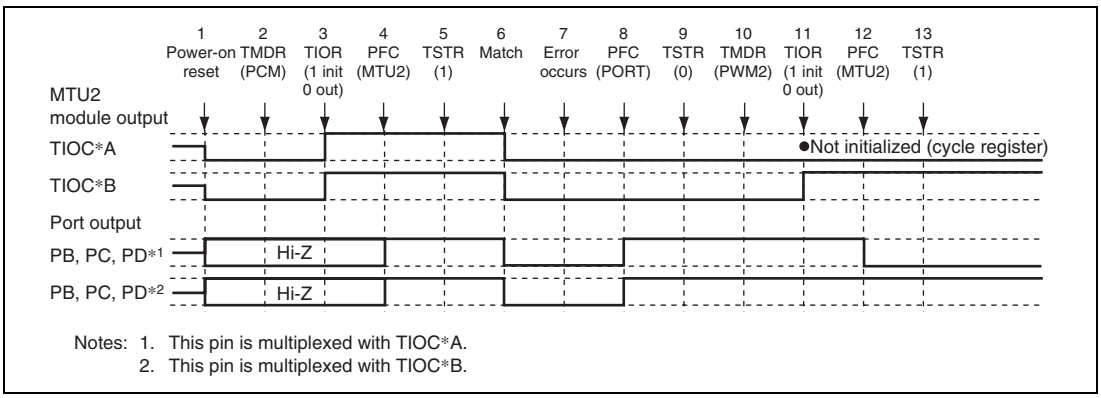


Figure 12.145 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

1 to 9 are the same as in figure 12.143.

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 12.146 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.

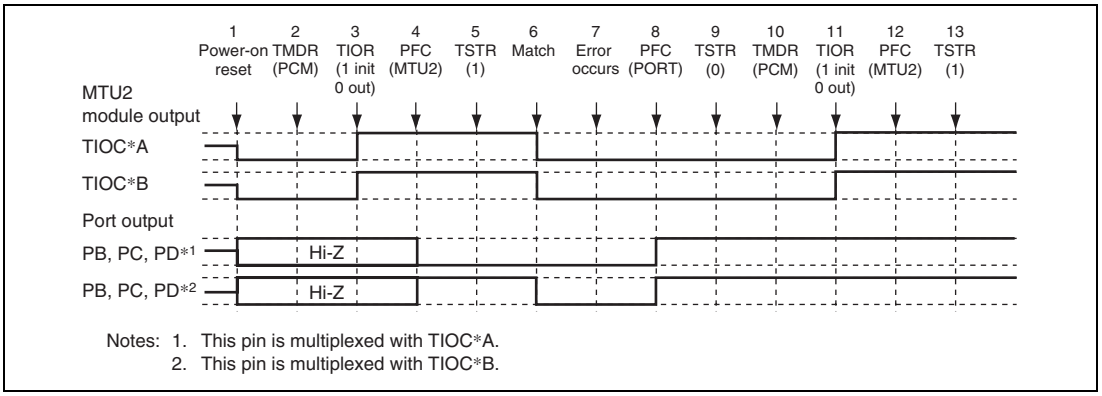


Figure 12.146 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

1 to 9 are the same as in figure 12.143.

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 12.147 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.

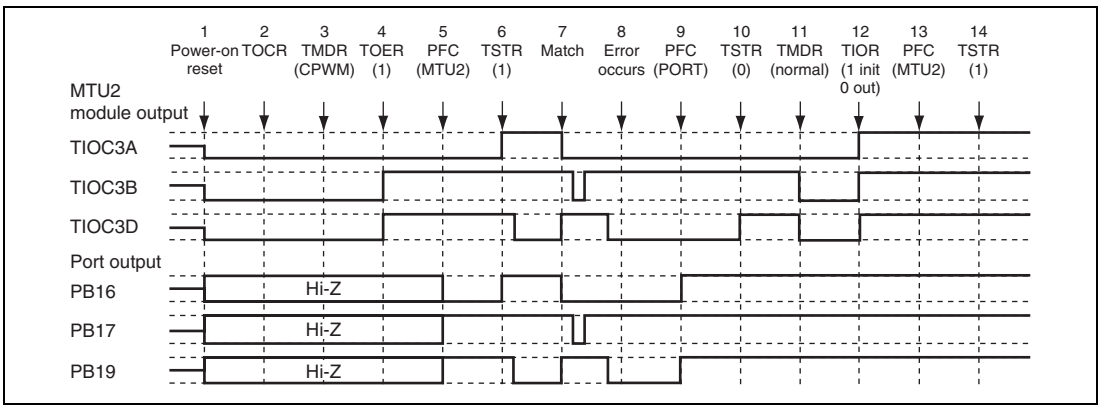


Figure 12.147 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

1. After a power-on 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 12.148 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.

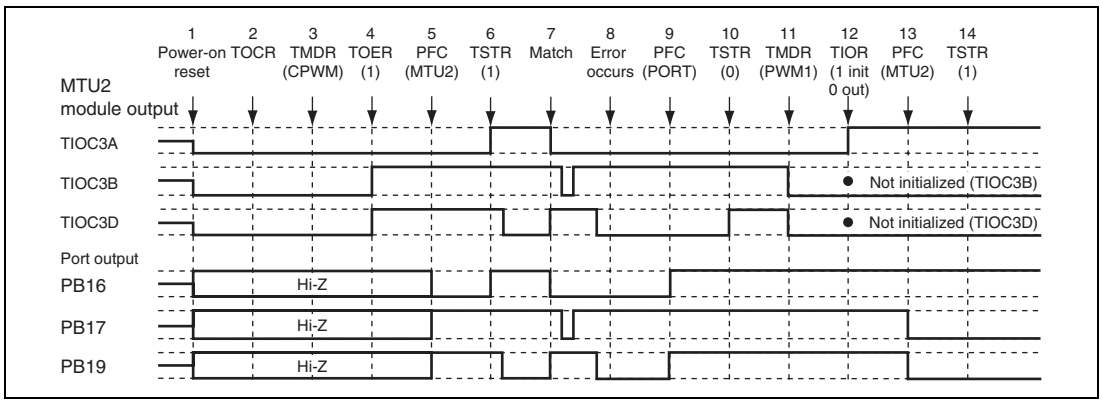


Figure 12.148 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.147.

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 12.149 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).

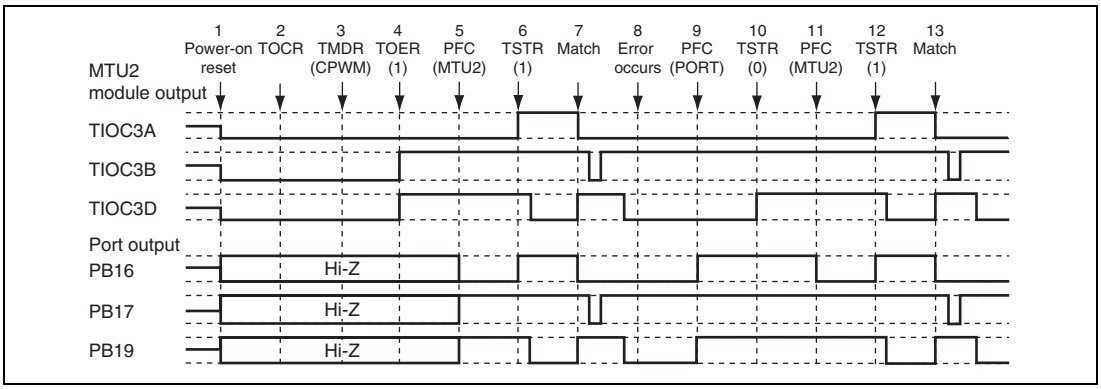


Figure 12.149 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.147.

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 12.150 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).

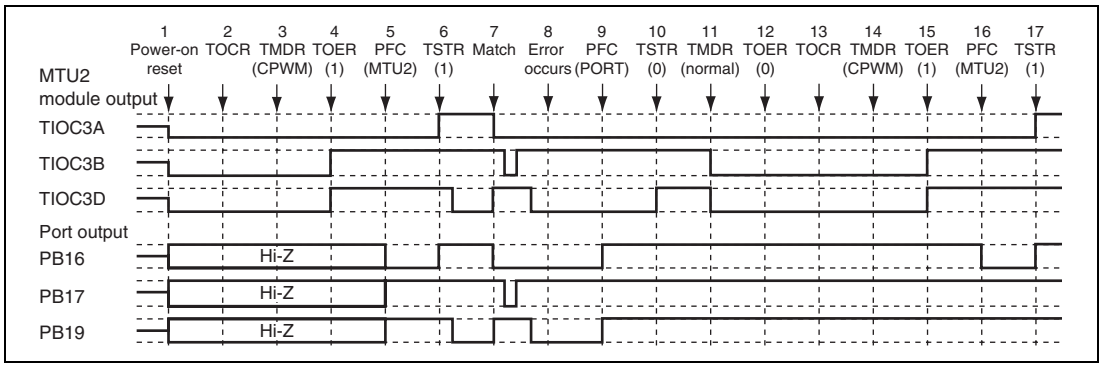


Figure 12.150 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.147.

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

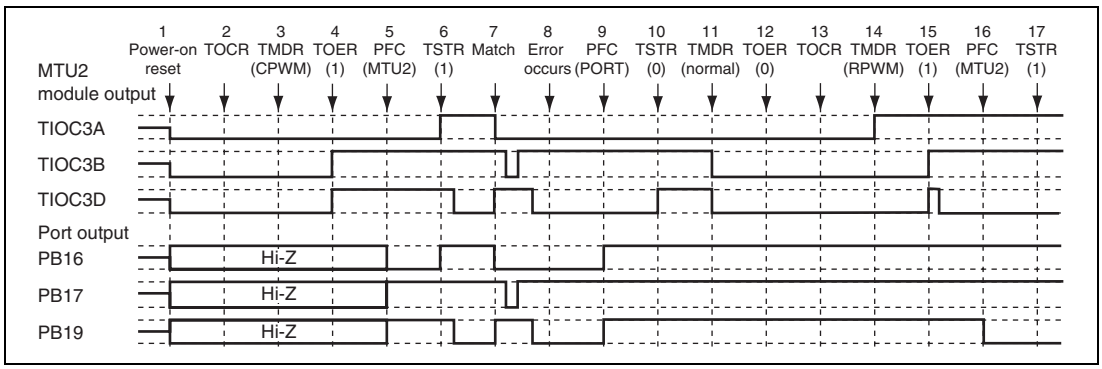


Figure 12.151 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 12.147.

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 12.152 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.

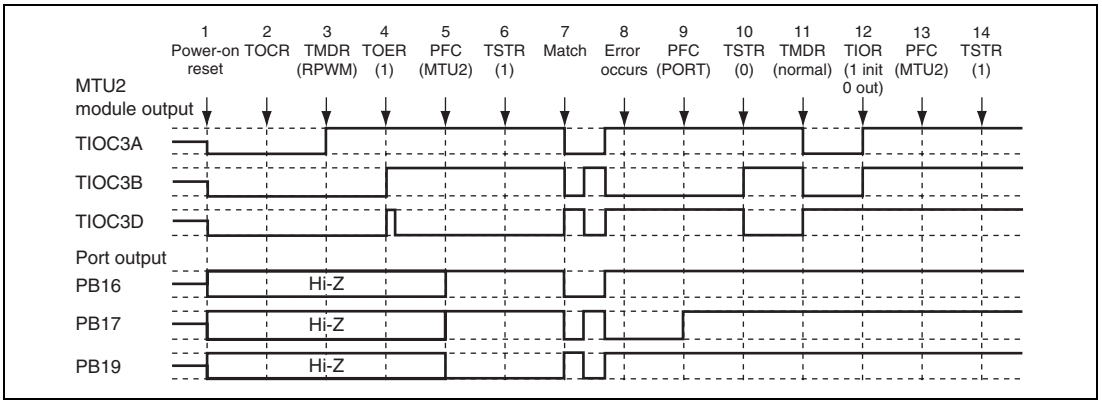


Figure 12.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

1. After a power-on 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 12.153 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.

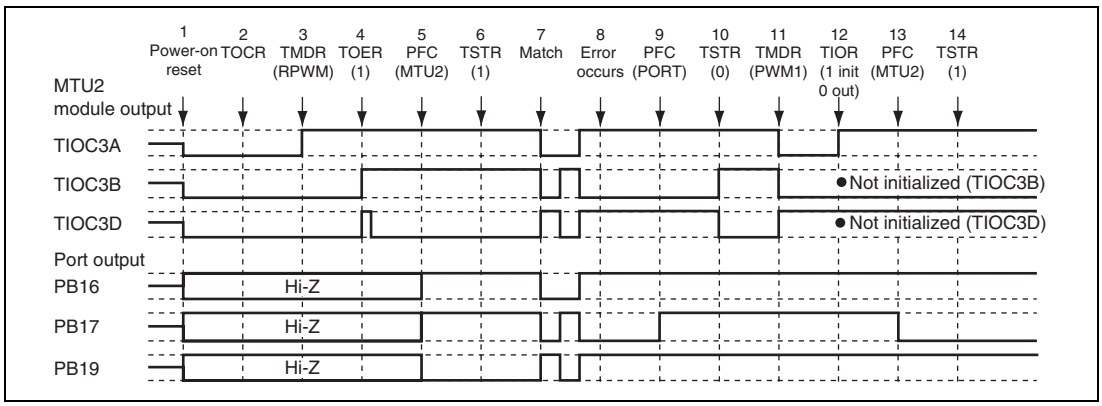


Figure 12.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

1 to 10 are the same as in figure 12.152.

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 12.154 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.

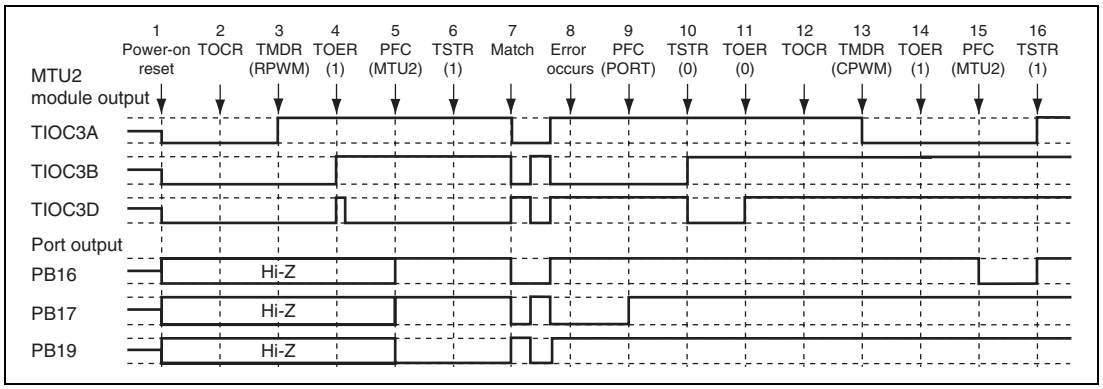


Figure 12.154 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

1 to 10 are the same as in figure 12.152.

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 12.155 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.

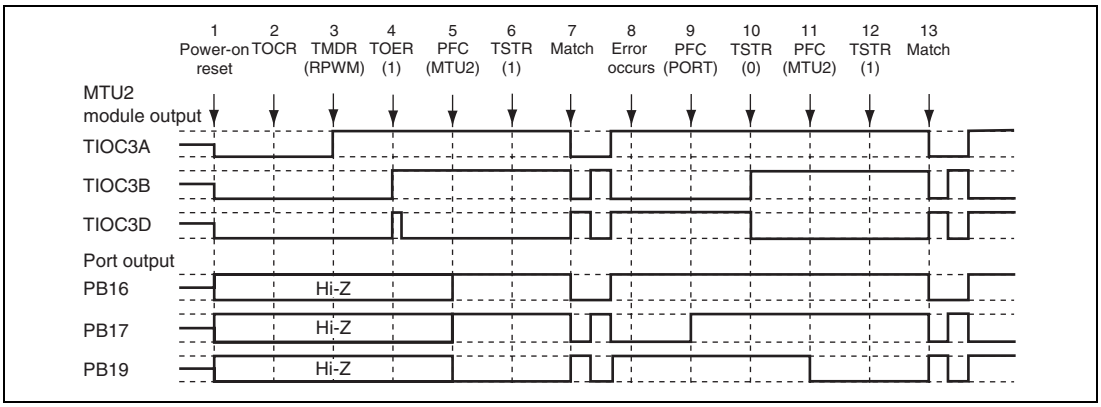


Figure 12.155 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

1 to 10 are the same as in figure 12.152.

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 13 8-Bit Timers (TMR)

This LSI has an on-chip 2-channel 8-bit timer based on an 8-bit counter. It can be used to count external events and, using compare-match signals with two registers, as a multifunction timer in a variety of applications, such as the generation of counter resets, interrupt requests, and pulse output with a user-defined duty cycle.

Figure 13.1 shows a block diagram of the 8-bit timer.

13.1 Features

- Selection of seven clock sources

  The counters can be driven by one of six internal clock signals (Pϕ/8, Pϕ/64, Pϕ/8192, Pϕ/2, Pϕ/32, or Pϕ/1024) or an external clock input.
- Selection of three ways to clear the counters

  The counters can be cleared on compare match A or B, or by an external reset signal.
- Timer output control by a combination of two compare match signals

  The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to output pulses with a desired duty cycle or PWM output.
- Cascading of two channels (TMR_0 and TMR_1)

  Operation as a 16-bit timer is possible, using TMR_0 for the upper 8 bits and TMR_1 for the lower 8 bits (16-bit count mode).

  TMR_1 can be used to count TMR_0 compare matches (compare match count mode).
- Three interrupt sources

  Compare match A, compare match B, and overflow interrupts can be requested independently.
- Generation of trigger to start A/D converter conversion

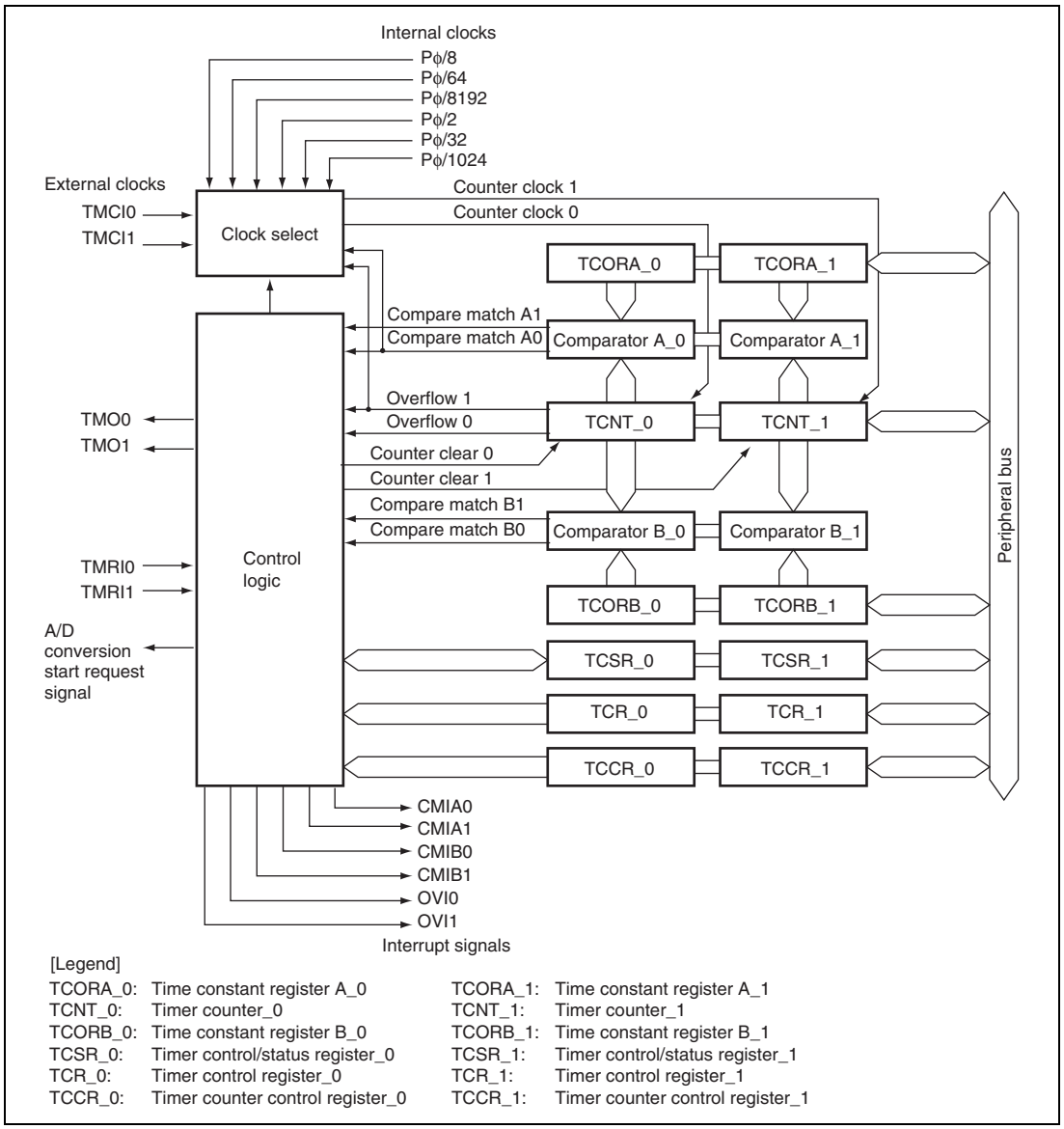


Figure 13.1 Block Diagram of 8-Bit Timer

13.2 Input/Output Pins

Table 13.1 shows the pin configuration of the TMR.

Table 13.1 Pin Configuration

Channel	Name	Symbol	I/O	Function
0	Timer output pin	TMO0	Output	Outputs compare match
	Timer clock input pin	TMCI0	Input	Inputs external clock for counter
	Timer reset input pin	TMRI0	Input	Inputs external reset to counter
1	Timer output pin	TMO1	Output	Outputs compare match
	Timer clock input pin	TMCI1	Input	Inputs external clock for counter
	Timer reset input pin	TMRI1	Input	Inputs external reset to counter

13.3 Register Descriptions

The TMR has the following registers.

Channel 0:

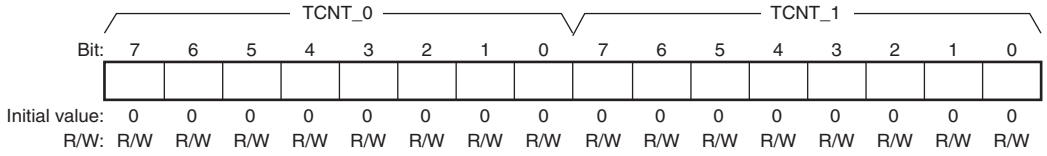
- Timer counter_0 (TCNT_0)
- Time constant register A_0 (TCORA_0)
- Time constant register B_0 (TCORB_0)
- Timer control register_0 (TCR_0)
- Timer counter control register_0 (TCCR_0)
- Timer control/status register_0 (TCSR_0)

Channel 1:

- Timer counter_1 (TCNT_1)
- Time constant register A_1 (TCORA_1)
- Time constant register B_1 (TCORB_1)
- Timer control register_1 (TCR_1)
- Timer counter control register_1 (TCCR_1)
- Timer control/status register_1 (TCSR_1)

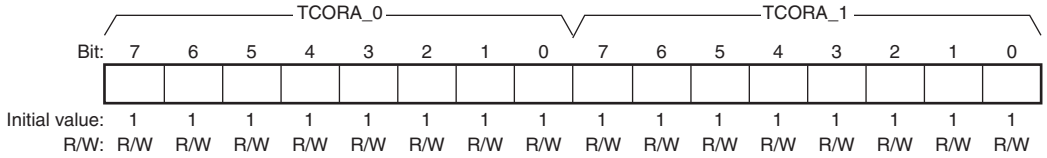
13.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT_0 and TCNT_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. Bits CKS2 to CKS0 in TCR and bits ICKS1 and ICKS0 in TCCR are used to select a clock. TCNT can be cleared by an external reset input signal, compare match A signal, or compare match B signal. Which signal is to be used for clearing is selected by bits CCLR1 and CCLR0 in TCR. When TCNT overflows from H'FF to H'00, bit OVF in TCSR is set to 1. TCNT is initialized to H'00.



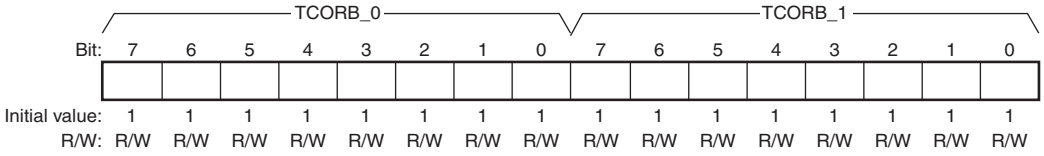
13.3.2 Time Constant Register A (TCORA)

TCORA is an 8-bit readable/writable register. TCORA_0 and TCORA_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note however that comparison is disabled during the T2 state of a TCORA write cycle. The timer output from the TMO pin can be freely controlled by this compare match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR. TCORA is initialized to H'FF.



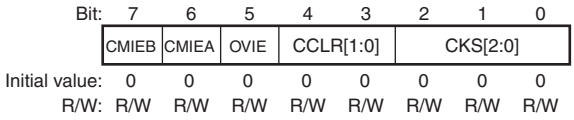
13.3.3 Time Constant Register B (TCORB)

TCORB is an 8-bit readable/writable register. TCORB_0 and TCORB_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set to 1. Note however that comparison is disabled during the T2 state of a TCORB write cycle. The timer output from the TMO pin can be freely controlled by this compare match signal (compare match B) and the settings of bits OS3 and OS2 in TCSR. TCORB is initialized to H'FF.



13.3.4 Timer Control Register (TCR)

TCR selects the TCNT clock source and the condition for clearing TCNT, and enables/disables interrupt requests.



Bit	Bit Name	Initial Value	R/W	Description
7	CMIEB	0	R/W	Compare Match Interrupt Enable B Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag in TCSR is set to 1. 0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled
6	CMIEA	0	R/W	Compare Match Interrupt Enable A Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1. 0: CMFA interrupt requests (CMIA) are disabled 1: CMFA interrupt requests (CMIA) are enabled

Bit	Bit Name	Initial Value	R/W	Description
5	OVIE	0	R/W	Timer Overflow Interrupt Enable Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1. 0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled
4, 3	CCLR[1:0]	00	R/W	Counter Clear 1 and 0* These bits select the method by which TCNT is cleared. 00: Clearing is disabled 01: Cleared by compare match A 10: Cleared by compare match B 11: Cleared at rising edge (TMRIS in TCCR is cleared to 0) of the external reset input or when the external reset input is high (TMRIS in TCCR is set to 1)
2 to 0	CKS[2:0]	000	R/W	Clock Select 2 to 0* These bits select the clock input to TCNT and count condition. See table 13.2.

Note: * To use an external reset or external clock, the function of the corresponding pin should be selected using the pin function controller (PFC). For details, see section 23, Pin Function Controller (PFC).

13.3.5 Timer Counter Control Register (TCCR)

TCCR selects the TCNT internal clock source and controls external reset input.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	TMRIS	—	ICKS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R/W	Reserved These bits are always read as 0. The write value should always be 0.
3	TMRIS	0	R/W	Timer Reset Input Select Selects an external reset input when the CCLR1 and CCLR0 bits in TCR are B'11. 0: Cleared at rising edge of the external reset 1: Cleared when the external reset is high
2	—	0	R/W	Reserved This bit is always read as 0. The write value should always be 0
1, 0	ICKS[1:0]	00	R/W	Internal Clock Select 1 and 0 These bits in combination with bits CKS2 to CKS0 in TCR select the internal clock. See table 13.2.

Table 13.2 Clock Input to TCNT and Count Condition

	TCR			TCCR		
Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Bit 1 ICKS1	Bit 0 ICKS0	Description
TMR_0	0	0	0	—	—	Clock input prohibited.
	0	0	1	0	0	Uses internal clock. Counts at rising edge of Pϕ/8.
				0	1	Uses internal clock. Counts at rising edge of Pϕ/2.
				1	0	Uses internal clock. Counts at falling edge of Pϕ/8.
				1	1	Uses internal clock. Counts at falling edge of Pϕ/2.
	0	1	0	0	0	Uses internal clock. Counts at rising edge of Pϕ/64.
				0	1	Uses internal clock. Counts at rising edge of Pϕ/32.
				1	0	Uses internal clock. Counts at falling edge of Pϕ/64.
				1	1	Uses internal clock. Counts at falling edge of Pϕ/32.
	0	1	1	0	0	Uses internal clock. Counts at rising edge of Pϕ/8192.
				0	1	Uses internal clock. Counts at rising edge of Pϕ/1024.
				1	0	Uses internal clock. Counts at falling edge of Pϕ/8192.
				1	1	Uses internal clock. Counts at falling edge of Pϕ/1024.
	1	0	0	—	—	Counts at TCNT_1 overflow signal*[1].
TMR_1	0	0	0	—	—	Clock input prohibited.
	0	0	1	0	0	Uses internal clock. Counts at rising edge of Pϕ/8.
				0	1	Uses internal clock. Counts at rising edge of Pϕ/2.
				1	0	Uses internal clock. Counts at falling edge of Pϕ/8.
				1	1	Uses internal clock. Counts at falling edge of Pϕ/2.
	0	1	0	0	0	Uses internal clock. Counts at rising edge of Pϕ/64.
				0	1	Uses internal clock. Counts at rising edge of Pϕ/32.
				1	0	Uses internal clock. Counts at falling edge of Pϕ/64.
				1	1	Uses internal clock. Counts at falling edge of Pϕ/32.
	0	1	1	0	0	Uses internal clock. Counts at rising edge of Pϕ/8192.
				0	1	Uses internal clock. Counts at rising edge of Pϕ/1024.
				1	0	Uses internal clock. Counts at falling edge of Pϕ/8192.
				1	1	Uses internal clock. Counts at falling edge of Pϕ/1024.
	1	0	0	—	—	Counts at TCNT_0 compare match A*[1].

	TCR			TCCR		
Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Bit 1 ICKS1	Bit 0 ICKS0	Description
All	1	0	1	—	—	Uses external clock. Counts at rising edge*[2].
	1	1	0	—	—	Uses external clock. Counts at falling edge*[2].
	1	1	1	—	—	Uses external clock. Counts at both rising and falling edges*[2].

Notes: 1. If the clock input of TMR_0 is the TCNT_1 overflow signal and that of TMR_1 is the TCNT_0 compare match signal, no incrementing clock is generated. Do not use this setting.
2. To use the external clock, the function of the corresponding pin should be selected using the pin function controller (PFC). For details, see section 23, Pin Function Controller (PFC).

13.3.6 Timer Control/Status Register (TCSR)

TCSR displays status flags, and controls compare match output.

- TCSR_0

Bit:	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OS[3:2]		OS[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

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

- TCSR_1

Bit:	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS[3:2]		OS[1:0]	
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R	R/W	R/W	R/W	R/W

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

- TCSR_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*1	Compare Match Flag B [Setting condition] • When TCNT matches TCORB [Clearing condition] • When writing 0 after reading CMFB = 1
6	CMFA	0	R/(W)*1	Compare Match Flag A [Setting condition] • When TCNT matches TCORA [Clearing condition] • When writing 0 after reading CMFA = 1
5	OVF	0	R/(W)*1	Timer Overflow Flag [Setting condition] • When TCNT overflows from H'FF to H'00 [Clearing condition] • When writing 0 after reading OVF = 1
4	ADTE	0	R/W	A/D Trigger Enable Selects enabling or disabling of A/D converter start requests by compare match A. 0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are enabled
3, 2	OS[3:2]	00	R/W	Output Select 3 and 2*2 These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)

Bit	Bit Name	Initial Value	R/W	Description
1, 0	OS[1:0]	00	R/W	Output Select 1 and 0*[2] These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Notes: 1. Only 0 can be written to bits 7 to 5, to clear these flags.
2. Timer output is disabled when bits OS3 to OS0 are all 0. Timer output is 0 until the first compare match occurs after resetting.

- TCSR_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*[1]	Compare Match Flag B [Setting condition] • When TCNT matches TCORB [Clearing condition] • When writing 0 after reading CMFB = 1
6	CMFA	0	R/(W)*[1]	Compare Match Flag A [Setting condition] • When TCNT matches TCORA [Clearing condition] • When writing 0 after reading CMFA = 1
5	OVF	0	R/(W)*[1]	Timer Overflow Flag [Setting condition] • When TCNT overflows from H'FF to H'00 [Clearing condition] • When writing 0 after reading OVF = 1
4	—	0	R	Reserved This is a read-only bit and cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	OS[3:2]	00	R/W	Output Select 3 and 2*[2] These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)
1, 0	OS[1:0]	00	R/W	Output Select 1 and 0*[2] These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Notes: 1. Only 0 can be written to bits 7 to 5, to clear these flags.
2. Timer output is disabled when bits OS3 to OS0 are all 0. Timer output is 0 until the first compare match occurs after resetting.

13.4 Operation

13.4.1 Pulse Output

Figure 13.2 shows an example of the 8-bit timer being used to generate a pulse output with a desired duty cycle. The control bits are set as follows:

1. In TCR, clear bit CCLR1 to 0 and set bit CCLR0 to 1 so that TCNT is cleared at a TCORA compare match.
2. In TCSR, set bits OS3 to OS0 to B'0110, causing the output to change to 1 at a TCORA compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required. The output level of the 8-bit timer holds 0 until the first compare match occurs after a reset.

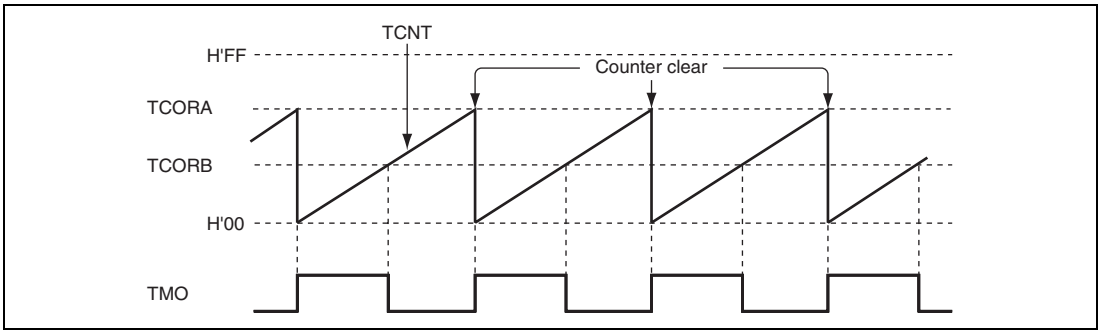


Figure 13.2 Example of Pulse Output

13.4.2 Reset Input

Figure 13.3 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRI input. The control bits are set as follows:

1. Set both bits CCLR1 and CCLR0 in TCR to 1 and set the TMRIS bit in TCCR to 1 so that TCNT is cleared at the high level input of the TMRI signal.
2. In TCSR, set bits OS3 to OS0 to B'0110, causing the output to change to 1 at a TCORA compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRI input determined by TCORA and with a pulse width determined by TCORB and TCORA.

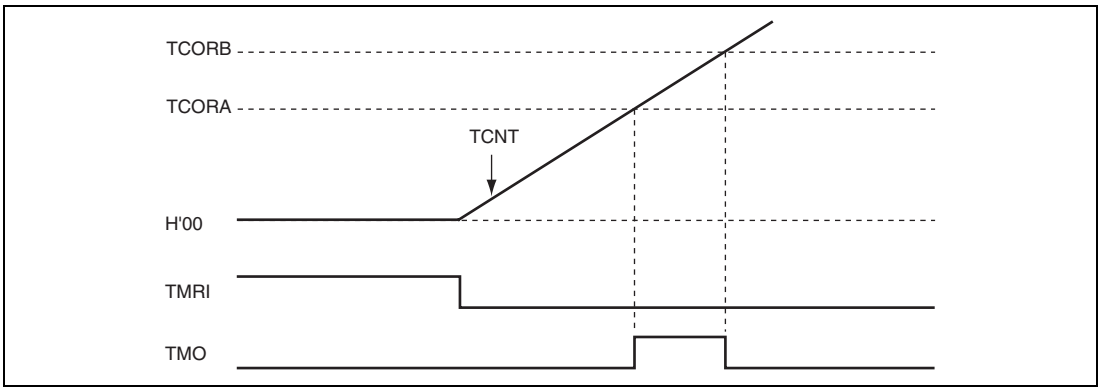


Figure 13.3 Example of Reset Input

13.5 Operation Timing

13.5.1 TCNT Count Timing

Figure 13.4 shows the TCNT count timing for internal clock input. Figure 13.5 shows the TCNT count timing for external clock input. Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

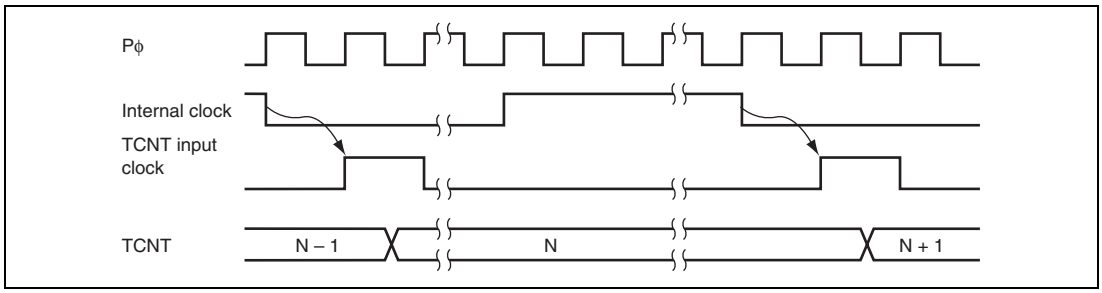


Figure 13.4 Count Timing for Internal Clock Input at Falling Edge

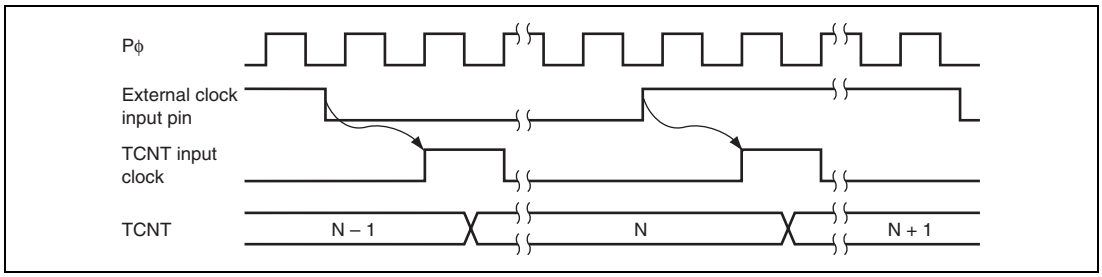


Figure 13.5 Count Timing for External Clock Input at Falling and Rising Edges

13.5.2 Timing of CMFA and CMFB Setting at Compare Match

The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT values match. The compare match signal is generated at the last state in which the match is true, just before the timer counter is updated. Therefore, when the TCOR and TCNT values match, the compare match signal is not generated until the next TCNT clock input. Figure 13.6 shows this timing.

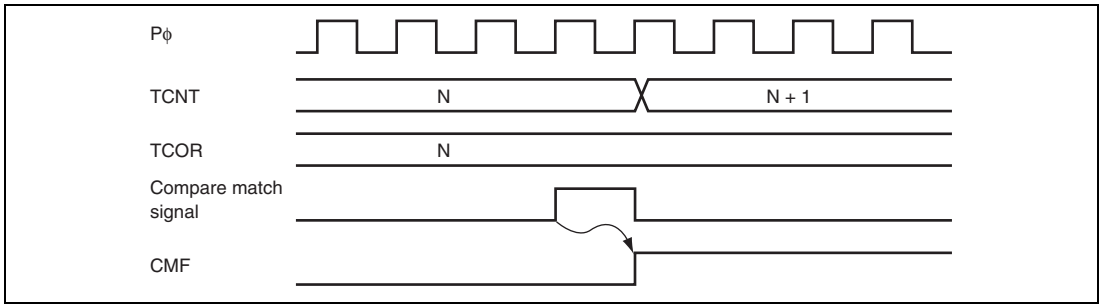


Figure 13.6 Timing of CMF Setting at Compare Match

13.5.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the timer output changes as specified by bits OS3 to OS0 in TCSR. Figure 13.7 shows the timing when the timer output is toggled by the compare match A signal.

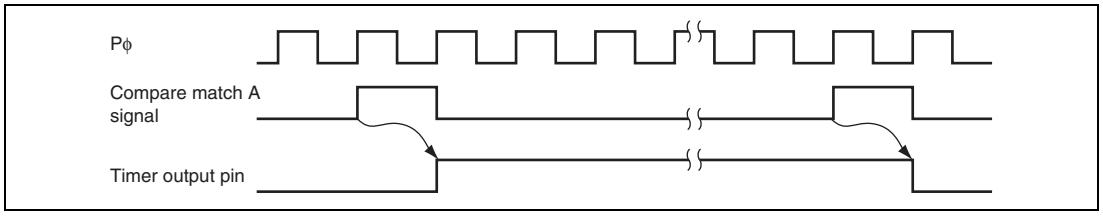


Figure 13.7 Timing of Toggled Timer Output at Compare Match A

13.5.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of bits CCLR1 and CCLR0 in TCR. Figure 13.8 shows the timing of this operation.

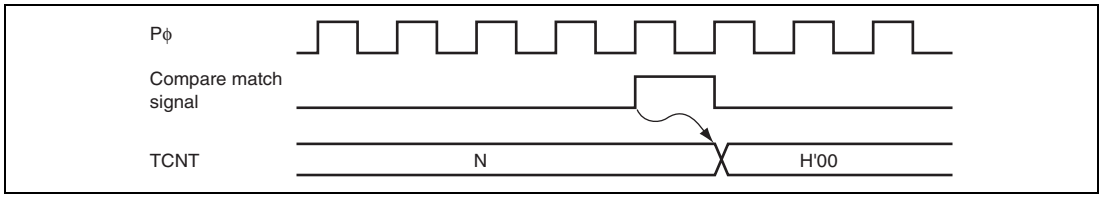


Figure 13.8 Timing of Counter Clear by Compare Match

13.5.5 Timing of TCNT External Reset

TCNT is cleared at the rising edge or high level of an external reset input, depending on the settings of bits CCLR1 and CCLR0 in TCR. The clear pulse width must be at least 2 states. Figures 13.9 and 13.10 show the timing of this operation.

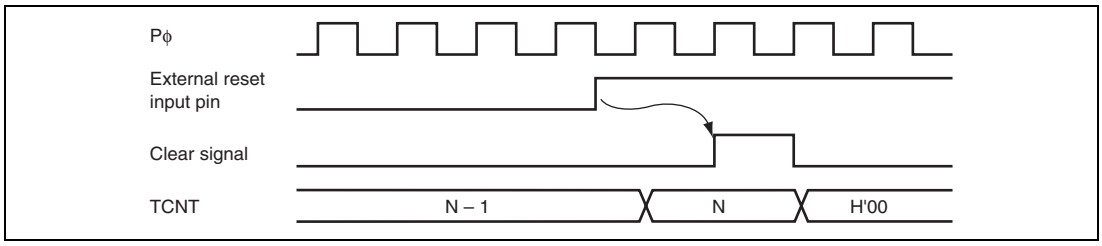


Figure 13.9 Timing of Clearance by External Reset (Rising Edge)

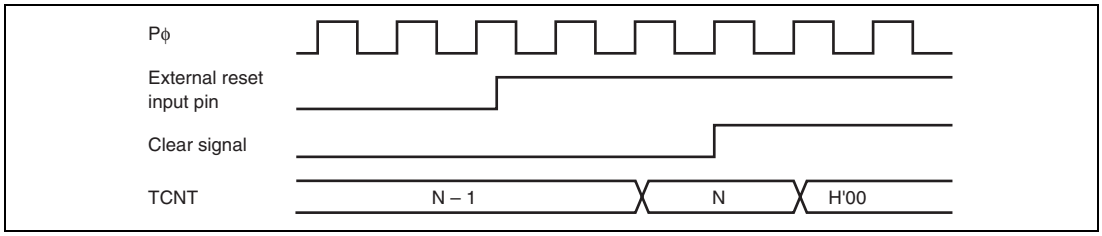


Figure 13.10 Timing of Clearance by External Reset (High Level)

13.5.6 Timing of Overflow Flag (OVF) Setting

The OVF bit in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Figure 13.11 shows the timing of this operation.

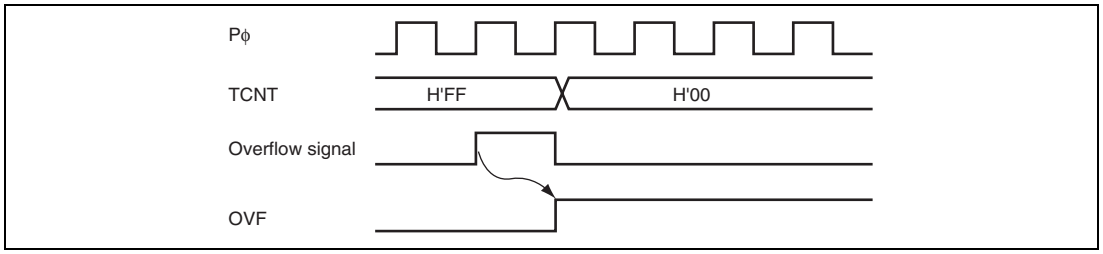


Figure 13.11 Timing of OVF Setting

13.6 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR_0 or TCR_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match count mode).

13.6.1 16-Bit Counter Mode

When bits CKS2 to CKS0 in TCR_0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

(1) Setting of Compare Match Flags

- The CMF flag in TCSR_0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR_1 is set to 1 when a lower 8-bit compare match event occurs.

(2) Counter Clear Specification

- If the CCLR1 and CCLR0 bits in TCR_0 have been set for counter clear at compare match, the 16-bit counter (TCNT_0 and TCNT_1 together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter clear by the TMRI0 pin has been set.
- The settings of the CCLR1 and CCLR0 bits in TCR_1 are ignored. The lower 8 bits cannot be cleared independently.

(3) Pin Output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR_0 is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR_1 is in accordance with the lower 8-bit compare match conditions.

13.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR_1 are set to B'100, TCNT_1 counts compare match A for channel 0. Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

13.7 Interrupt Sources

13.7.1 Interrupt Sources

There are three interrupt sources for the 8-bit timer (TMR_0 or TMR_1): CMIA, CMIB, and OVI. Their interrupt sources and priorities are shown in table 13.3. Each interrupt source is enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent interrupt requests are sent for each to the interrupt controller.

Table 13.3 8-Bit Timer (TMR_0 or TMR_1) Interrupt Sources

Name	Interrupt Source	Interrupt Flag	Priority
CMIA0	TCORA_0 compare match	CMFA	High
CMIB0	TCORB_0 compare match	CMFB	↑
OVI0	TCNT_0 overflow	OVF	Low
CMIA1	TCORA_1 compare match	CMFA	High
CMIB1	TCORB_1 compare match	CMFB	↑
OVI1	TCNT_1 overflow	OVF	Low

13.7.2 A/D Converter Activation

The A/D converter can be activated only by TMR_0 compare match A.

If the ADTE bit in TCSR_0 is set to 1 when the CMFA flag in TCSR_0 is set to 1 by the occurrence of TMR_0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

13.8 Usage Notes

13.8.1 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last state in the cycle in which the values of TCNT and TCOR match. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula.

$$f = P\phi/(N + 1)$$

f: Counter frequency
Pϕ: Operating frequency
N: TCOR value

13.8.2 Conflict between TCNT Write and Clear

If a counter clear signal is generated during the T_2 state of a TCNT write cycle, the clear takes priority and the write is not performed as shown in figure 13.12.

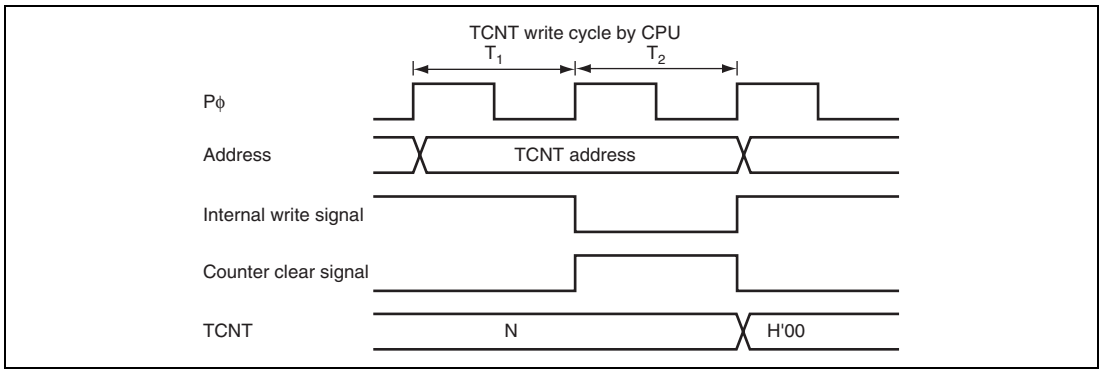


Figure 13.12 Conflict between TCNT Write and Clear

13.8.3 Conflict between TCNT Write and Increment

If a TCNT input clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the counter is not incremented as shown in figure 13.13.

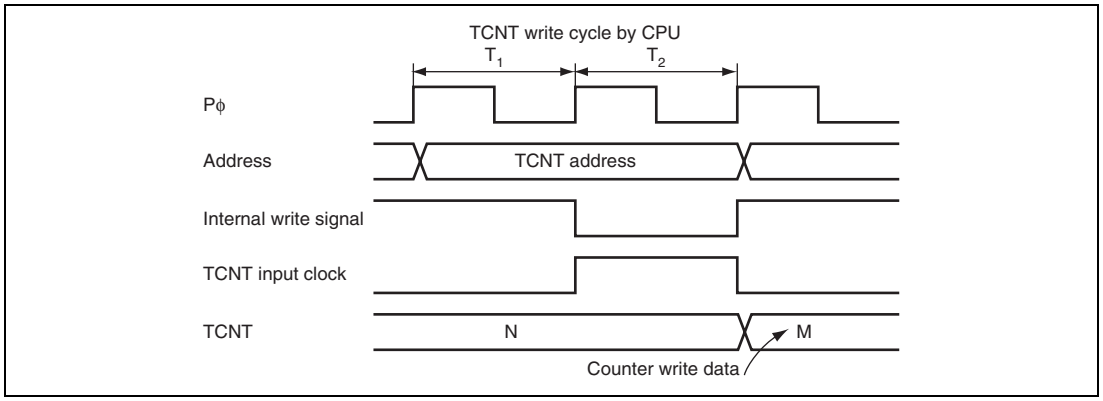


Figure 13.13 Conflict between TCNT Write and Increment

13.8.4 Conflict between TCOR Write and Compare Match

If a compare match event occurs during the T_2 state of a TCOR write cycle, the TCOR write takes priority and the compare match signal is inhibited as shown in figure 13.14.

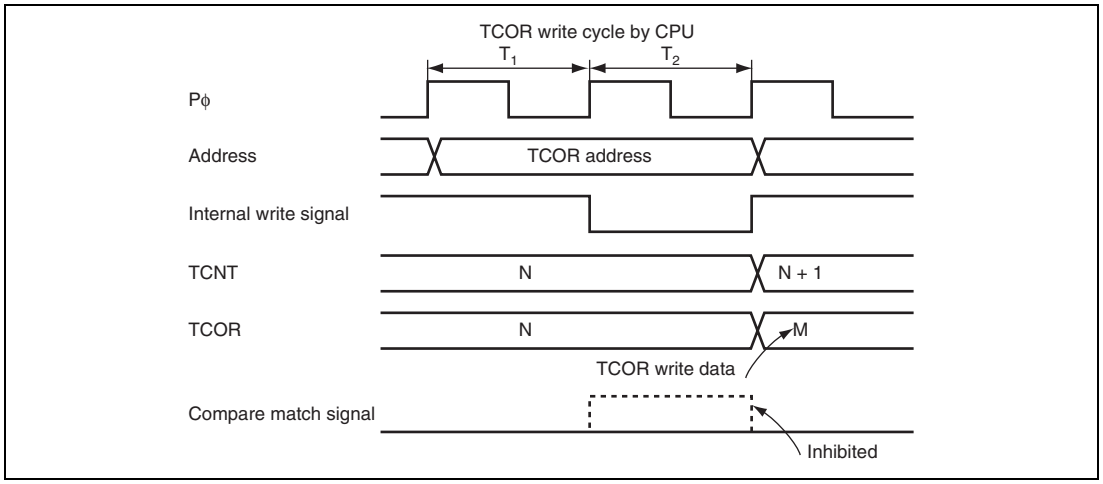


Figure 13.14 Conflict between TCOR Write and Compare Match

13.8.5 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 13.4.

Table 13.4 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1-output	↑
0-output	
No change	Low

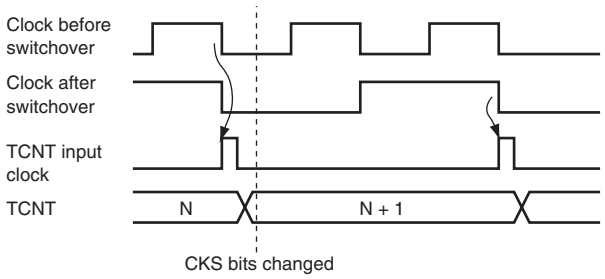
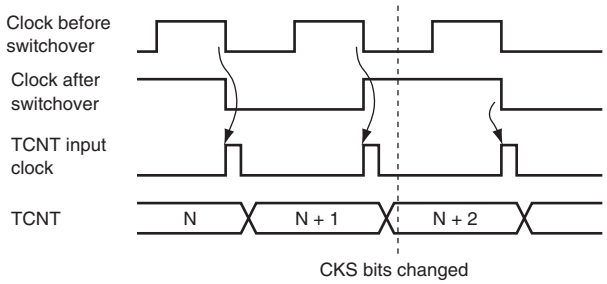
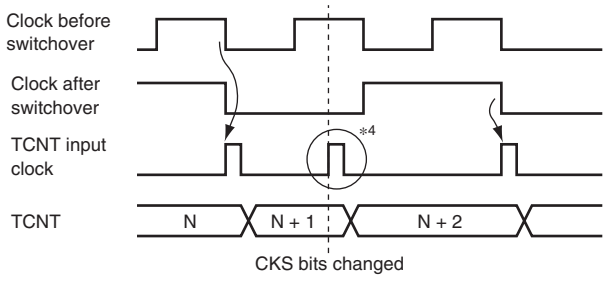
13.8.6 Switching of Internal Clocks and TCNT Operation

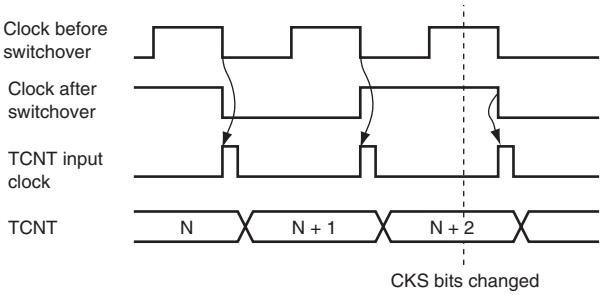
TCNT may be incremented erroneously depending on when the internal clock is switched. Table 13.5 shows the relationship between the timing at which the internal clock is switched (by writing to bits CKS1 and CKS0) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the rising or falling edge of the internal clock pulse are always monitored. Table 13.5 assumes that the falling edge is selected. If the signal levels of the clocks before and after switching change from high to low as shown in item 3, the change is considered as the falling edge. Therefore, a TCNT clock pulse is generated and TCNT is incremented. This is similar to when the rising edge is selected.

The erroneous incrementation of TCNT can also happen when switching between rising and falling edges of the internal clock, and when switching between internal and external clocks.

Table 13.5 Switching of Internal Clock and TCNT Operation

No.	Timing to Change CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	
3	Switching from high to low*3	

No.	Timing to Change CKS1 and CKS0 Bits	TCNT Clock Operation
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated because the change of the signal levels is considered as a falling edge; TCNT is incremented.

13.8.7 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TCNT_0 and TCNT_1 are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

13.8.8 Module Standby Setting

Operation of the TMR can be disabled or enabled using the standby control register. The initial setting is for operation of the TMR to be halted. Register access is enabled by clearing module standby mode. For details, see section 25, Power-Down Modes.

13.8.9 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. Interrupts should therefore be disabled before entering module standby mode.

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.

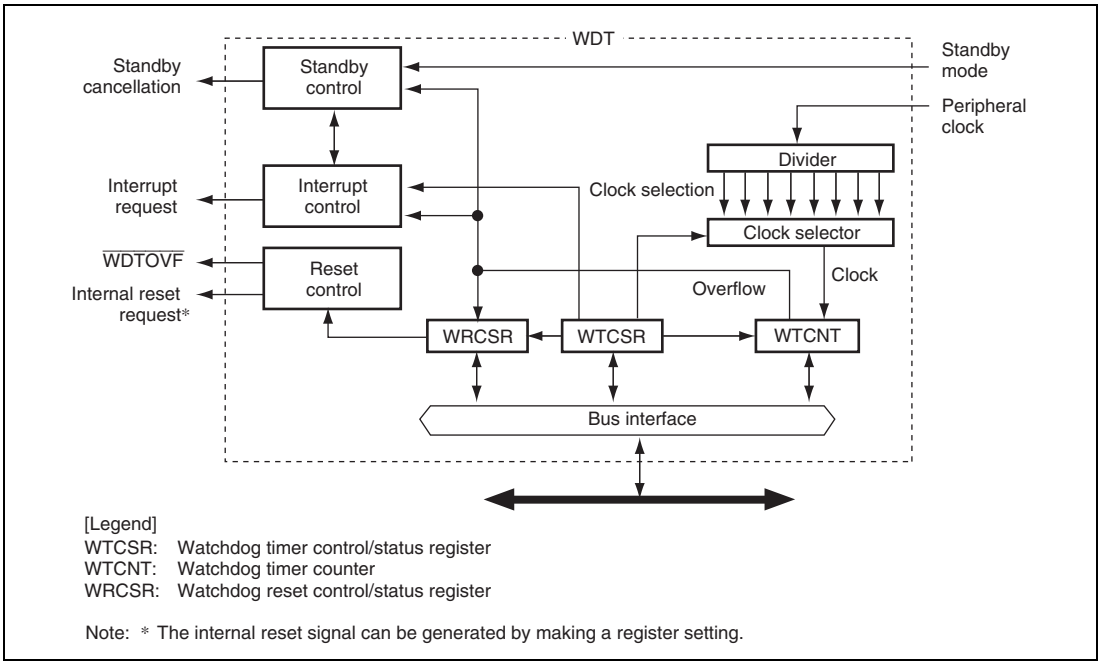


Figure 14.1 Block Diagram of WDT

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 deep standby mode or 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.

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

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 deep standby mode or 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 mode 1: Use as watchdog timer mode 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 inside the parenthesis in the table is the value when the peripheral clock (Pϕ) is 25 MHz.

Bits 2 to 0	Clock Ratio	Overflow Cycle
000:	1 × Pϕ	10.2 μs
001:	1/64 × Pϕ	655.4 μs
010:	1/128 × Pϕ	1.3 ms
011:	1/256 × Pϕ	2.6 ms
100:	1/512 × Pϕ	5.2 ms
101:	1/1024 × Pϕ	10.5 ms
110:	1/4096 × Pϕ	41.9 ms
111:	1/16384 × Pϕ	167.8 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 or in deep standby mode, 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.

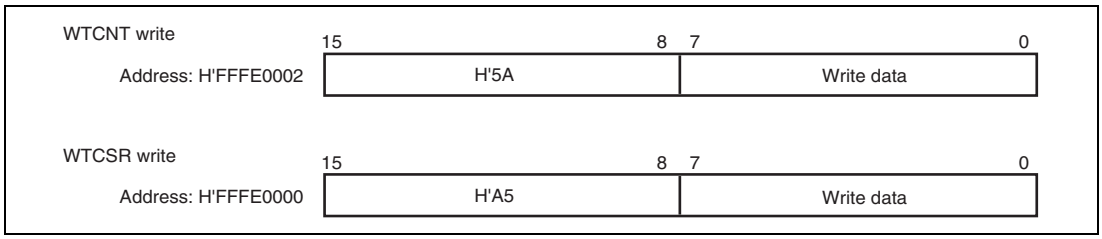


Figure 14.2 Writing to WTCNT and WTCSR

(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.

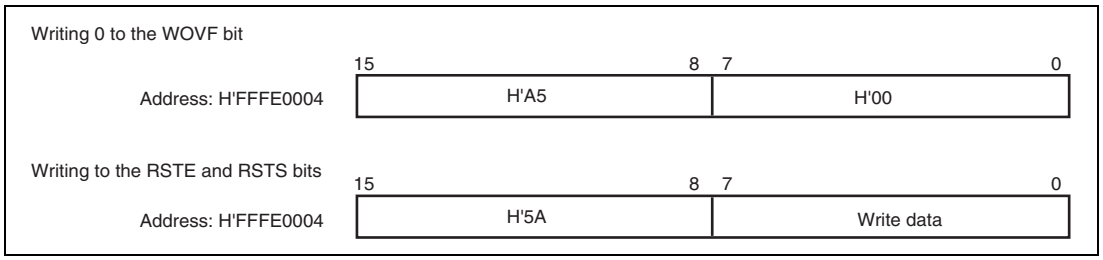


Figure 14.3 Writing to WRCSR

(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 to 1 and the DEEP bit to 0 in the standby control register (STBCR: see section 25, Power-Down Modes), the execution of a SLEEP instruction places 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. Note that, the WDT counts up by the clock to be set.
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.

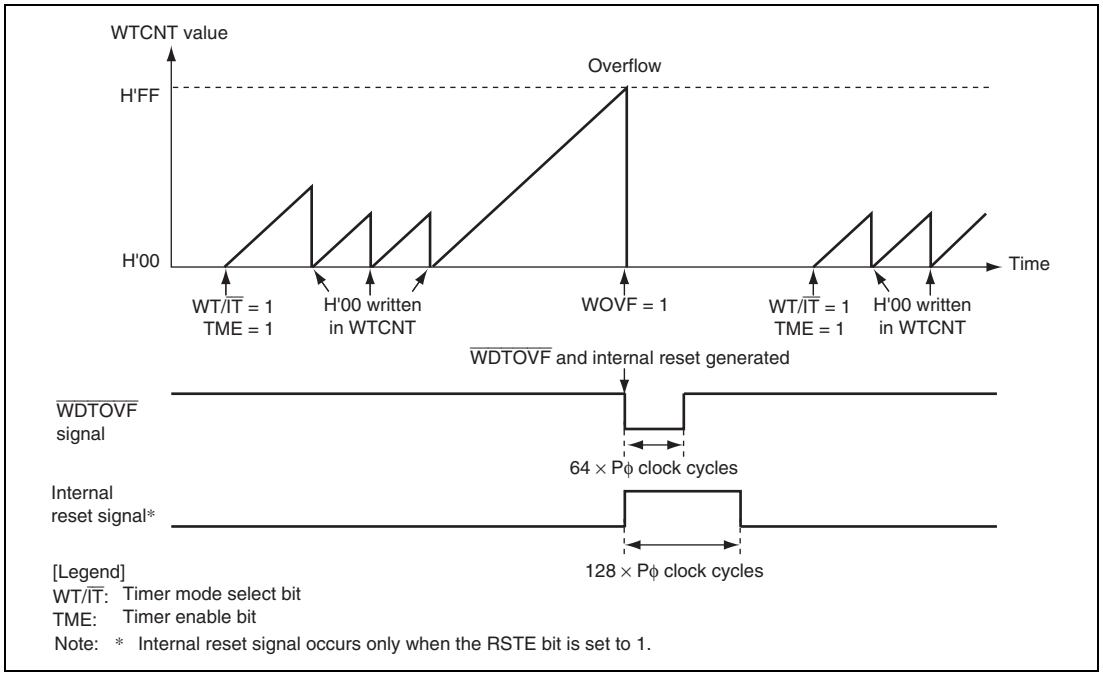


Figure 14.4 Operation in Watchdog Timer Mode

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.

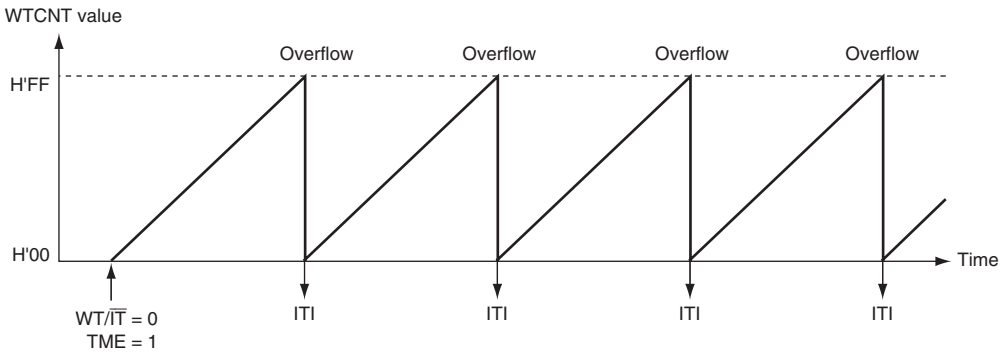


Figure 14.5 Operation in Interval Timer Mode

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 Interval Timer Overflow Flag

When the value in WTCNT is H'FF, the IOVF flag in WTCSR cannot be cleared.

Only clear the IOVF flag when the value in WTCNT has either become H'00 or been changed to a value other than H'FF.

14.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ signal is input to the $\overline{\text{RES}}$ pin of this LSI, this LSI cannot be initialized correctly.

Avoid input of the $\overline{\text{WDTOVF}}$ signal to the $\overline{\text{RES}}$ pin of this LSI through glue logic circuits. To reset the entire system with the $\overline{\text{WDTOVF}}$ signal, use the circuit shown in figure 14.6.

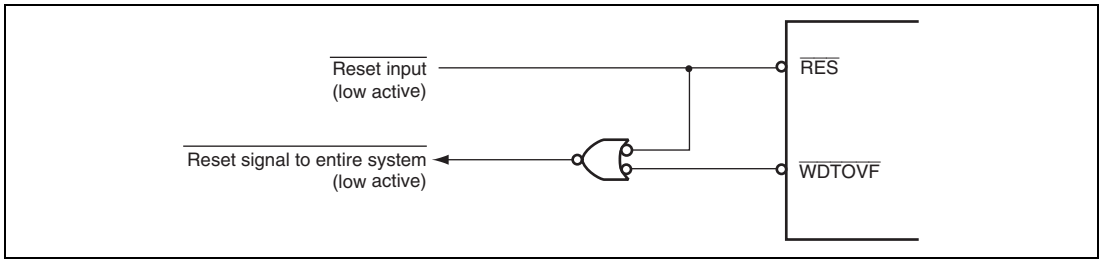


Figure 14.6 Example of System Reset Circuit Using $\overline{\text{WDTOVF}}$ Signal

14.5.5 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 during DMAC burst transfer, manual reset exception handling will be pended until the CPU acquires the bus mastership.

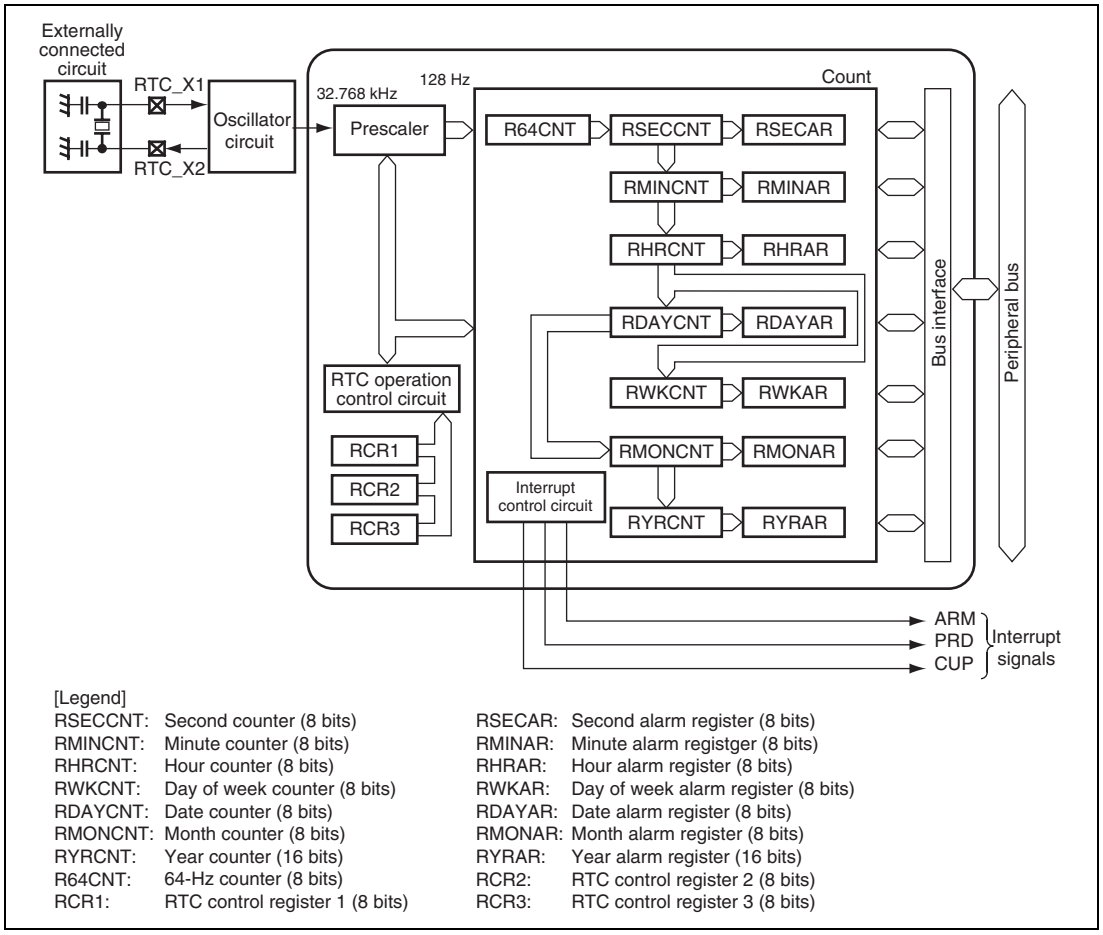
Section 15 Realtime Clock (RTC)

This LSI has a realtime clock (RTC) with its own 32.768-kHz crystal oscillator.

15.1 Features

- Clock and calendar functions (BCD format): Seconds, minutes, hours, date, day of the week, month, and year
- 1-Hz to 64-Hz timer (binary format)
  64-Hz counter indicates the state of the RTC divider circuit between 64 Hz and 1 Hz
- Start/stop function
- 30-second adjust function
- Alarm interrupt: Frame comparison of seconds, minutes, hours, date, day of the week, month, and year can be used as conditions for the alarm interrupt
- Periodic interrupts: the interrupt cycle may be 1/256 second, 1/64 second, 1/16 second, 1/4 second, 1/2 second, 1 second, or 2 seconds
- Carry interrupt: a carry interrupt indicates when a carry occurs during a counter read
- Automatic leap year adjustment

Figure 15.1 shows the block diagram of RTC.



[Legend]

RSECCNT:	Second counter (8 bits)	RSECAR:	Second alarm register (8 bits)
RMINCNT:	Minute counter (8 bits)	RMINAR:	Minute alarm registger (8 bits)
RHRCNT:	Hour counter (8 bits)	RHRAR:	Hour alarm register (8 bits)
RWKCNT:	Day of week counter (8 bits)	RWKAR:	Day of week alarm register (8 bits)
RDAYCNT:	Date counter (8 bits)	RDAYAR:	Date alarm register (8 bits)
RMONCNT:	Month counter (8 bits)	RMONAR:	Month alarm register (8 bits)
RYRCNT:	Year counter (16 bits)	RYRAR:	Year alarm register (16 bits)
R64CNT:	64-Hz counter (8 bits)	RCR2:	RTC control register 2 (8 bits)
RCR1:	RTC control register 1 (8 bits)	RCR3:	RTC control register 3 (8 bits)

Figure 15.1 RTC Block Diagram

15.2 Input/Output Pin

Table 15.1 shows the RTC pin configuration.

Table 15.1 Pin Configuration

Name	Abbreviation	I/O	Description
RTC oscillator crystal pin	RTC_X1	Input	Connects 32.768-kHz crystal resonator for RTC
RTC oscillator crystal pin	RTC_X2	Output	Connects 32.768-kHz crystal resonator for RTC

15.3 Register Descriptions

The RTC has the following registers.

Table 15.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
64-Hz counter	R64CNT	R	H'xx	H'FFFE0800	8
Second counter	RSECCNT	R/W	H'xx	H'FFFE0802	8
Minute counter	RMINCNT	R/W	H'xx	H'FFFE0804	8
Hour counter	RHRCNT	R/W	H'xx	H'FFFE0806	8
Day of week counter	RWKCNT	R/W	H'0x	H'FFFE0808	8
Date counter	RDAYCNT	R/W	H'xx	H'FFFE080A	8
Month counter	RMONCNT	R/W	H'xx	H'FFFE080C	8
Year counter	RYRCNT	R/W	H'xxxx	H'FFFE080E	16
Second alarm register	RSECAR	R/W	H'xx	H'FFFE0810	8
Minute alarm register	RMINAR	R/W	H'xx	H'FFFE0812	8
Hour alarm register	RHRAR	R/W	H'xx	H'FFFE0814	8
Day of week alarm register	RWKAR	R/W	H'0x	H'FFFE0816	8
Date alarm register	RDAYAR	R/W	H'xx	H'FFFE0818	8
Month alarm register	RMONAR	R/W	H'xx	H'FFFE081A	8
Year alarm register	RYRAR	R/W	H'xxxx	H'FFFE0820	16

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
RTC control register 1	RCR1	R/W	H'00	H'FFFE081C	8
RTC control register 2	RCR2	R/W	H'09	H'FFFE081E	8
RTC control register 3	RCR3	R/W	H'00	H'FFFE0824	8

15.3.1 64-Hz Counter (R64CNT)

R64CNT indicates the state of the divider circuit between 64 Hz and 1 Hz.

Reading this register, when carry from 128-Hz divider stage is generated, sets the CF bit in the RTC control register 1 (RCR1) to 1 so that the carrying and reading 64 Hz counter are performed at the same time is indicated. In this case, the R64CNT should be read again after writing 0 to the CF bit in RCR1 since the read value is not valid.

After the RESET bit or ADJ bit in the RTC control register 2 (RCR2) is set to 1, the RTC divider circuit is initialized and R64CNT is initialized to H'00.

R64CNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
Initial value:	0	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

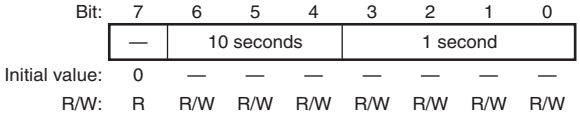
Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	1 Hz	Undefined	R	Indicate the state of the divider circuit between 64 Hz and 1 Hz.
5	2 Hz	Undefined	R	
4	4 Hz	Undefined	R	
3	8 Hz	Undefined	R	
2	16 Hz	Undefined	R	
1	32 Hz	Undefined	R	
0	64 Hz	Undefined	R	

15.3.2 Second Counter (RSECCNT)

RSECCNT is used for setting/counting in the BCD-coded second section. The count operation is performed by a carry for each second of the 64-Hz counter.

The assignable range is from 00 through 59 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RSECCNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.



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	10 seconds	Undefined	R/W	Counting Ten's Position of Seconds Counts on 0 to 5 for 60-seconds counting.
3 to 0	1 second	Undefined	R/W	Counting One's Position of Seconds Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

15.3.3 Minute Counter (RMINCNT)

RMINCNT is used for setting/counting in the BCD-coded minute section. The count operation is performed by a carry for each minute of the second counter.

The assignable range is from 00 through 59 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RMINCNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	—	10 minutes			1 minute			
Initial value:	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 to 4	10 minutes	Undefined	R/W	Counting Ten's Position of Minutes Counts on 0 to 5 for 60-minutes counting.
3 to 0	1 minute	Undefined	R/W	Counting One's Position of Minutes Counts on 0 to 9 once per second. When a carry is generated, 1 is added to the ten's position.

15.3.4 Hour Counter (RHRCNT)

RHRCNT is used for setting/counting in the BCD-coded hour section. The count operation is performed by a carry for each 1 hour of the minute counter.

The assignable range is from 00 through 23 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RHRCNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	—	—	10 hours		1 hour			
Initial value:	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, 4	10 hours	Undefined	R/W	Counting Ten's Position of Hours Counts on 0 to 2 for ten's position of hours.
3 to 0	1 hour	Undefined	R/W	Counting One's Position of Hours Counts on 0 to 9 once per hour. When a carry is generated, 1 is added to the ten's position.

15.3.5 Day of Week Counter (RWKCNT)

RWKCNT is used for setting/counting day of week section. The count operation is performed by a carry for each day of the date counter.

The assignable range is from 0 through 6 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RWKCNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	Day		
Initial value:	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 to 0	Day	Undefined	R/W	Day-of-Week Counting Day-of-week is indicated with a binary code. 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

15.3.6 Date Counter (RDAYCNT)

RDAYCNT is used for setting/counting in the BCD-coded date section. The count operation is performed by a carry for each day of the hour counter.

The assignable range is from 01 through 31 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RDAYCNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

The range of date changes with each month and in leap years. Please confirm the correct setting. Leap years are recognized by dividing the year counter values by 400, 100, and 4 and obtaining a fractional result of 0. The year counter value of 0000 is included in the leap year.

Bit:	7	6	5	4	3	2	1	0
	—	—	10 days		1 day			
Initial value:	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, 4	10 days	Undefined	R/W	Counting Ten's Position of Dates
3 to 0	1 day	Undefined	R/W	Counting One's Position of Dates Counts on 0 to 9 once per date. When a carry is generated, 1 is added to the ten's position.

15.3.7 Month Counter (RMONCNT)

RMONCNT is used for setting/counting in the BCD-coded month section. The count operation is performed by a carry for each month of the date counter.

The assignable range is from 01 through 12 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RMONCNT is not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	10 months	1 month			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W

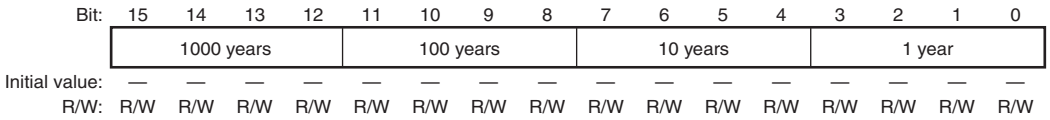
Bit	Bit Name	Initial Value	R/W	Description
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Counting Ten's Position of Months
3 to 0	1 month	Undefined	R/W	Counting One's Position of Months Counts on 0 to 9 once per month. When a carry is generated, 1 is added to the ten's position.

15.3.8 Year Counter (RYRCNT)

RYRCNT is used for setting/counting in the BCD-coded year section. The count operation is performed by a carry for each year of the month counter.

The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors occur. Carry out write processing after stopping the count operation through the setting of the START bit in RCR2.

RYRCNT is not initialized by a power-on reset or manual reset, in deep standby mode or software standby mode.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Counting Thousand's Position of Years
11 to 8	100 years	Undefined	R/W	Counting Hundred's Position of Years
7 to 4	10 years	Undefined	R/W	Counting Ten's Position of Years
3 to 0	1 year	Undefined	R/W	Counting One's Position of Years

15.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the BCD coded second counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.

The ENB bit in RSECAR is initialized to 0 by a power-on reset or in deep standby mode. The other bits are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	10 seconds			1 second			
Initial value:	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	ENB	0	R/W	When this bit is set to 1, a comparison with the RSECCNT value is performed.
6 to 4	10 seconds	Undefined	R/W	Ten's position of seconds setting value
3 to 0	1 second	Undefined	R/W	One's position of seconds setting value

15.3.10 Minute Alarm Register (RMINAR)

RMINAR is an alarm register corresponding to the minute counter RMINCNT. When the ENB bit is set to 1, a comparison with the RMINCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.

The ENB bit in RMINAR is initialized by a power-on reset or in deep standby mode. The other bits are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	10 minutes			1 minute			
Initial value:	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	ENB	0	R/W	When this bit is set to 1, a comparison with the RMINCNT value is performed.
6 to 4	10 minutes	Undefined	R/W	Ten's position of minutes setting value
3 to 0	1 minute	Undefined	R/W	One's position of minutes setting value

15.3.11 Hour Alarm Register (RHRAR)

RHRAR is an alarm register corresponding to the BCD coded hour counter RHRCNT of the RTC. When the ENB bit is set to 1, a comparison with the RHRCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 23 + ENB bits (practically in BCD), otherwise operation errors occur.

The ENB bit in RHRAR is initialized by a power-on reset or in deep standby mode. The other bits are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10 hours		1 hour			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RHRCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 hours	Undefined	R/W	Ten's position of hours setting value
3 to 0	1 hour	Undefined	R/W	One's position of hours setting value

15.3.12 Day of Week Alarm Register (RWKAR)

RWKAR is an alarm register corresponding to the BCD coded day of week counter RWKCNT. When the ENB bit is set to 1, a comparison with the RWKCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 0 through 6 + ENB bits (practically in BCD), otherwise operation errors occur.

The ENB bit in RWKAR is initialized by a power-on reset or in deep standby mode. The other bits are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	Day		
Initial value:	0	0	0	0	0	—	—	—
R/W:	R/W	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RWKCNT value is performed.
6 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	Day	Undefined	R/W	Day of week setting value 000: Sunday 001: Monday 010: Tuesday 011: Wednesday 100: Thursday 101: Friday 110: Saturday 111: Reserved (setting prohibited)

15.3.13 Date Alarm Register (RDAYAR)

RDAYAR is an alarm register corresponding to the BCD coded date counter RDAYCNT. When the ENB bit is set to 1, a comparison with the RDAYCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 01 through 31 + ENB bits (practically in BCD), otherwise operation errors occur.

The ENB bit in RDAYAR is initialized by a power-on reset or in deep standby mode. The other bits are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	10 days		1 day			
Initial value:	0	0	—	—	—	—	—	—
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RDAYCNT value is performed.
6	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
5, 4	10 days	Undefined	R/W	Ten's position of dates setting value
3 to 0	1 day	Undefined	R/W	One's position of dates setting value

15.3.14 Month Alarm Register (RMONAR)

RMONAR is an alarm register corresponding to the BCD coded month counter RMONCNT. When the ENB bit is set to 1, a comparison with the RMONCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 01 through 12 + ENB bits (practically in BCD), otherwise operation errors occur.

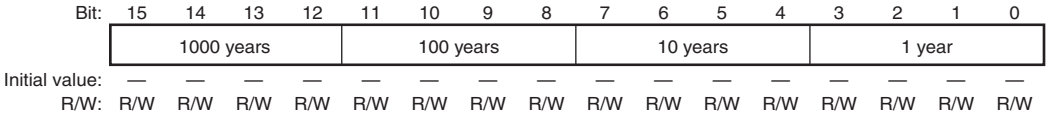
The ENB bit in RMONAR is initialized by a power-on reset or in deep standby mode. The other bits are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	10 months	1 month			
Initial value:	0	0	0	—	—	—	—	—
R/W:	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RMONCNT value is performed.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	10 months	Undefined	R/W	Ten's position of months setting value
3 to 0	1 month	Undefined	R/W	One's position of months setting value

15.3.15 Year Alarm Register (RYRAR)

RYRAR is an alarm register corresponding to the year counter RYRCNT. The assignable range is from 0000 through 9999 (practically in BCD), otherwise operation errors occur. RYRAR is not initialized by a power-on reset, a manual reset, or in deep standby mode and software standby mode.



Bit	Bit Name	Initial Value	R/W	Description
15 to 12	1000 years	Undefined	R/W	Thousand's position of years setting value
11 to 8	100 years	Undefined	R/W	Hundred's position of years setting value
7 to 4	10 years	Undefined	R/W	Ten's position of years setting value
3 to 0	1 year	Undefined	R/W	One's position of years setting value

15.3.16 RTC Control Register 1 (RCR1)

RCR1 is a register that affects carry flags and alarm flags. It also selects whether to generate interrupts for each flag.

RCR1 is initialized to H'00 by a power-on reset, a manual reset, or in deep standby mode. The CF flag is retained undefined until the division circuit is reset (the RESET and ADJ bits in RCR2 are set to 1). When using the CF flag, make sure to reset the divider circuit beforehand. This register is not initialized in software standby mode.

Bit:	7	6	5	4	3	2	1	0
	CF	—	—	CIE	AIE	—	—	AF
Initial value:	—	0	0	0	0	0	0	0
R/W:	R/W	R	R	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	CF	Undefined	R/W	Carry Flag Status flag that indicates that a carry has occurred. CF is set to 1 when a count-up to 64-Hz occurs at the second counter carry or 64-Hz counter read. A count register value read at this time cannot be guaranteed; another read is required. 0: No carry of 64-Hz counter by second counter or 64-Hz counter [Clearing condition] • When 0 is written to CF 1: Carry of 64-Hz counter by second counter or 64 Hz counter [Setting condition] • When the second counter or 64-Hz counter is read during a carry occurrence by the 64-Hz counter, or 1 is written to CF.
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	CIE	0	R/W	Carry Interrupt Enable Flag When the carry flag (CF) is set to 1, the CIE bit enables interrupts. 0: A carry interrupt is not generated when the CF flag is set to 1 1: A carry interrupt is generated when the CF flag is set to 1
3	AIE	0	R/W	Alarm Interrupt Enable Flag When the alarm flag (AF) is set to 1, the AIE bit allows interrupts. 0: An alarm interrupt is not generated when the AF flag is set to 1 1: An alarm interrupt is generated when the AF flag is set to 1
2, 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	AF	0	R/W	Alarm Flag The AF flag is set when the alarm time, which is set by an alarm register (ENB bit in RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, or RYRAR is set to 1), and counter match. 0: Alarm register and counter not match [Clearing condition] • When 0 is written to AF. 1: Alarm register and counter match* [Setting condition] • When alarm register (only a register with ENB bit set to 1) and counter match Note: * Writing 1 holds previous value.

15.3.17 RTC Control Register 2 (RCR2)

RCR2 is a register for periodic interrupt control, 30-second adjustment ADJ, divider circuit RESET, and RTC count control.

RCR2 is initialized to H'09 by a power-on reset or in deep standby mode. Bits other than the RTCEN and START bits are initialized by a manual reset. It is not initialized in software standby mode, and retains its contents.

Bit:	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
Initial value:	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	PEF	0	R/W	Periodic Interrupt Flag Indicates interrupt generation with the period designated by the PES2 to PES0 bits. When set to 1, PEF generates periodic interrupts. 0: Interrupts not generated with the period designated by the bits PES2 to PES0. [Clearing condition] • When 0 is written to PEF 1: Interrupts generated with the period designated by the PES2 to PES0 bits. [Setting condition] • When an interrupt is generated with the period designated by the bits PES0 to PES2 or when 1 is written to the PEF flag
6 to 4	PES[2:0]	000	R/W	Interrupt Enable Flags These bits specify the periodic interrupt. 000: No periodic interrupts generated 001: Periodic interrupt generated every 1/256 second 010: Periodic interrupt generated every 1/64 second 011: Periodic interrupt generated every 1/16 second 100: Periodic interrupt generated every 1/4 second 101: Periodic interrupt generated every 1/2 second 110: Periodic interrupt generated every 1 second 111: Periodic interrupt generated every 2 seconds

Bit	Bit Name	Initial Value	R/W	Description
3	RTCEN	1	R/W	Crystal Oscillator Control Controls the operation of the crystal oscillator for the RTC. 0: Halts the crystal oscillator for the RTC. 1: Runs the crystal oscillator for the RTC.
2	ADJ	0	R/W	30-Second Adjustment When 1 is written to the ADJ bit, times of 29 seconds or less will be rounded to 00 seconds and 30 seconds or more to 1 minute. The divider circuit (RTC prescaler and R64CNT) will be simultaneously reset. This bit always reads 0. Important: When using this bit, see section 15.5.5, Procedure for Setting the 30-Second Adjustment Function. 0: Runs normally. 1: 30-second adjustment.
1	RESET	0	R/W	Reset Writing 1 to this bit initializes the divider circuit. In this case, the RESET bit is automatically reset to 0 after 1 is written to and the divider circuit (RTC prescaler and R64CNT) is reset. Thus, there is no need to write 1 to this bit. This bit is always read as 0. 0: Runs normally. 1: Divider circuit is reset.
0	START	1	R/W	Start Bit Halts and restarts the counter (clock). 0: Second/minute/hour/day/week/month/year counter halts. 1: Second/minute/hour/day/week/month/year counter runs normally. Note: The 64-Hz counter always runs unless stopped with the RTCEN bit.

15.3.18 RTC Control Register 3 (RCR3)

When the ENB bit in RCR3 is set to 1, RCR3 compares the value of RYRCNT and that of RYRAR. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The ENB bit in RCR3 is initialized by a power-on reset or in deep standby mode. Remaining fields of RCR3 are not initialized by a power-on reset or manual reset, or in deep standby and software standby modes.

Bit:	7	6	5	4	3	2	1	0
	ENB	—	—	—	—	—	—	—
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	ENB	0	R/W	When this bit is set to 1, comparison of the year alarm register (RYRAR) and the year counter (RYRCNT) is performed.
6 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

15.4 Operation

RTC usage is shown below.

15.4.1 Initial Settings of Registers after Power-On

All the registers should be set after the power is turned on.

15.4.2 Setting Time

Figure 15.2 shows how to set the time when the clock is stopped.

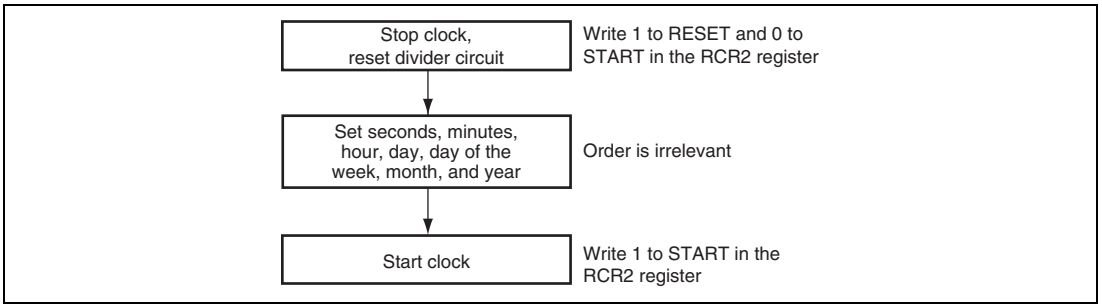


Figure 15.2 Setting Time

15.4.3 Reading Time

Figure 15.3 shows how to read the time.

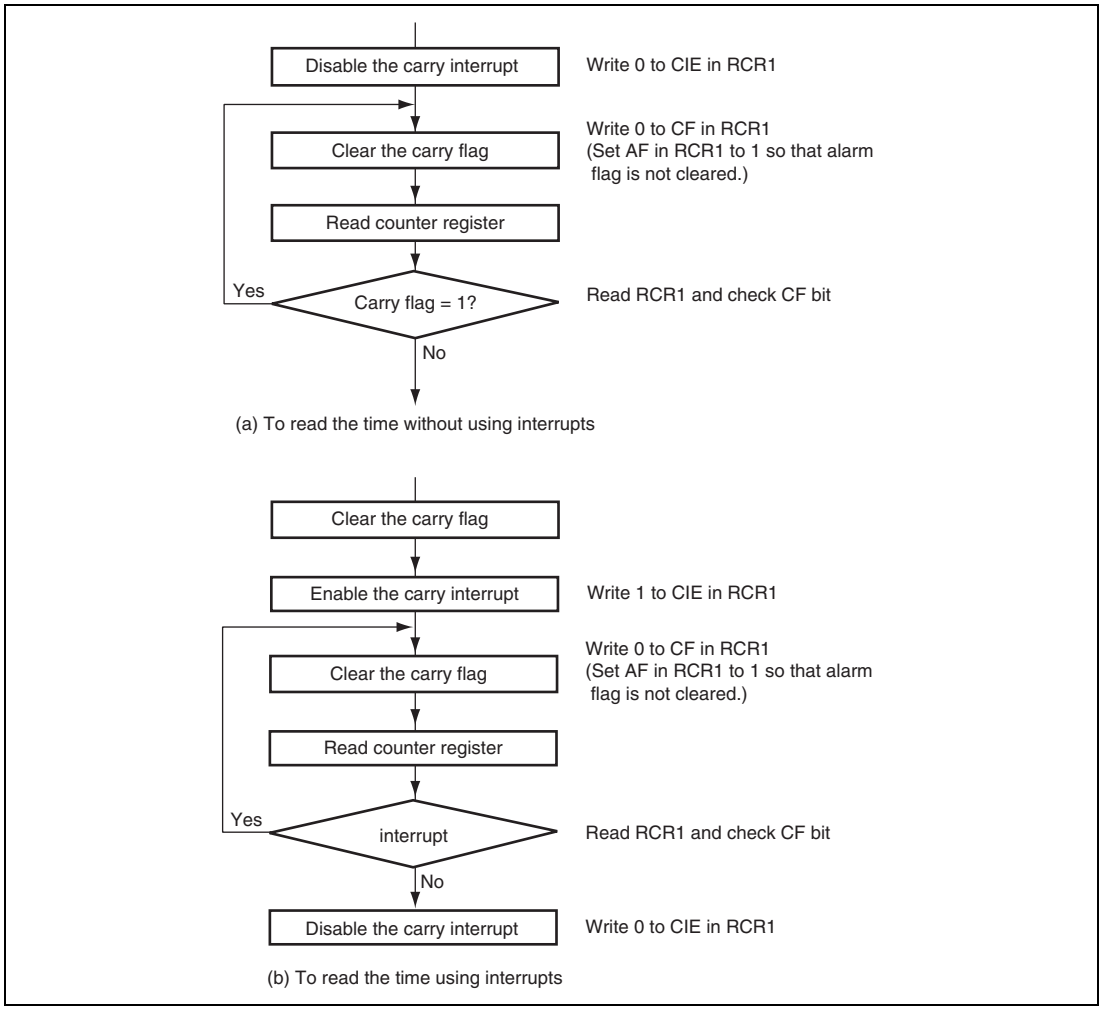


Figure 15.3 Reading Time

If a carry occurs while reading the time, the correct time will not be obtained, so it must be read again. Part (a) in figure 15.3 shows the method of reading the time without using interrupts; part (b) in figure 15.3 shows the method using carry interrupts. To keep programming simple, method (a) should normally be used.

15.4.4 Alarm Function

Figure 15.4 shows how to use the alarm function.

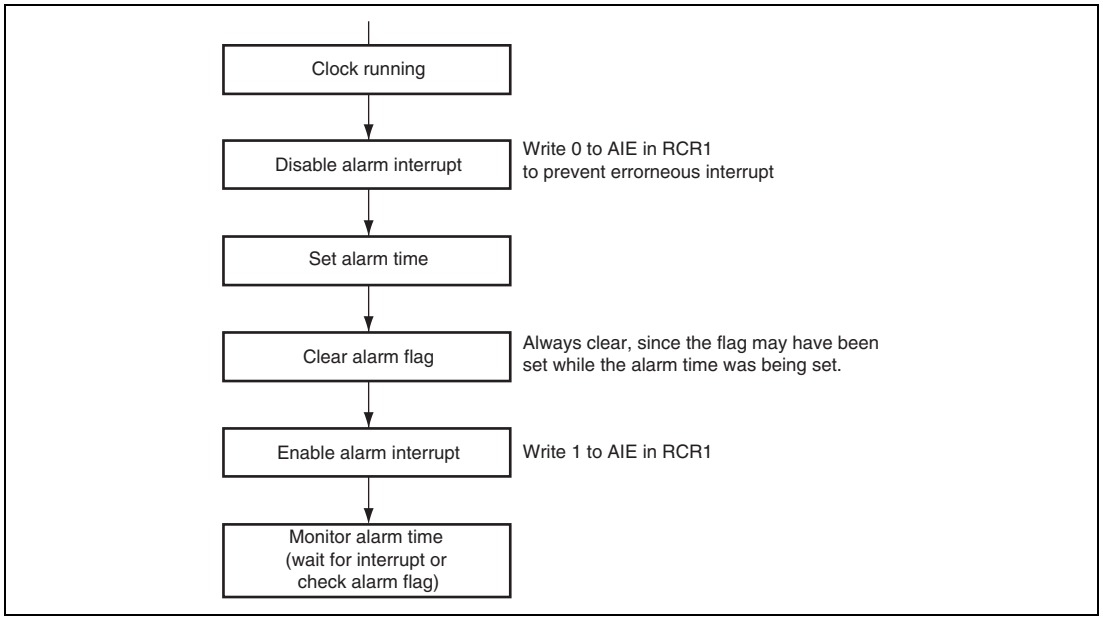


Figure 15.4 Using Alarm Function

Alarms can be generated using seconds, minutes, hours, day of the week, date, month, year, or any combination of these. Set the ENB bit in the register on which the alarm is placed to 1, and then set the alarm time in the lower bits. Clear the ENB bit in the register on which the alarm is not placed to 0.

When the clock and alarm times match, 1 is set in the AF bit in RCR1. Alarm detection can be checked by reading this bit, but normally it is done by interrupt. If 1 is set in the AIE bit in RCR1, an interrupt is generated when an alarm occurs.

The alarm flag is set when the clock and alarm times match. However, the alarm flag can be cleared by writing 0.

15.5 Usage Notes

15.5.1 Register Writing during RTC Count

Do not write to the count registers (RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, and RYRCNT) during the RTC counting (while the START bit in RCR2 is 1). If any of the count registers is written to during the RTC counting, the count register may not be read correctly immediately after the execution of a write instruction. The RTC counting must be stopped before writing to any of the count registers.

15.5.2 Use of Realtime Clock (RTC) Periodic Interrupts

The method of using the periodic interrupt function is shown in figure 15.5.

A periodic interrupt can be generated periodically at the interval set by the flags PES0 to PES2 in RCR2. When the time set by the PES0 to PES2 has elapsed, the PEF is set to 1.

The PEF is cleared to 0 upon periodic interrupt generation or when the flags PES0 to PES2 are set. Periodic interrupt generation can be confirmed by reading this bit, but normally the interrupt function is used.

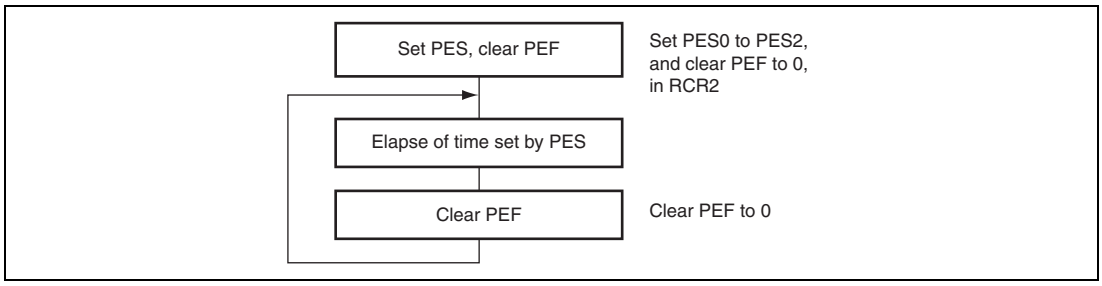


Figure 15.5 Using Periodic Interrupt Function

15.5.3 Transition to Standby Mode after Setting Register

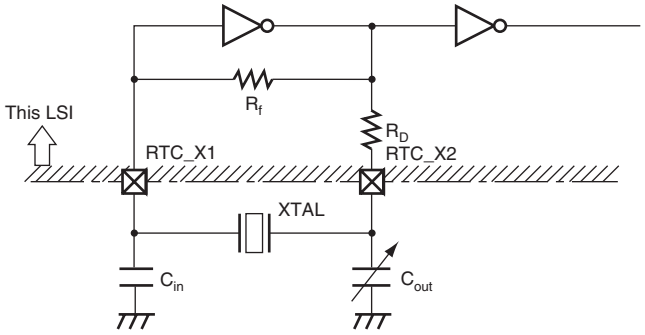
When a transition to standby mode is made after registers in the RTC are set, sometimes counting is not performed correctly. In case the registers are set, be sure to make a transition to standby mode after waiting for two count clocks or more.

15.5.4 Crystal Oscillator Circuit for RTC

Crystal oscillator circuit constants (recommended values) for the RTC are shown in table 15.3, and the RTC crystal oscillator circuit in figure 15.6.

Table 15.3 Crystal Oscillator Circuit Constants (Recommended Values)

f_{osc}	C_{in}	C_{out}
32.768 kHz	10 to 22 pF	10 to 22 pF



Notes: 1. Select either the C_{in} or C_{out} side for frequency adjustment variable capacitor according to requirements such as frequency range, degree of stability, etc.
2. Built-in resistance value R_f (Typ value) = 10 MΩ, R_D (Typ value) = 400 kΩ
3. C_{in} and C_{out} values include floating capacitance due to the wiring. Take care when using a ground plane.
4. The crystal oscillation stabilization time may differ depending on the mounted circuit component constants, stray capacitance, and so forth, so a suitable value should be determined in consultation with the resonator manufacturer.
5. Place the crystal resonator and load capacitors C_{in} and C_{out} as close as possible to the chip. Make wiring length as short as possible. Do not allocate signal lines close to oscillation circuit.
(Correct oscillation may not be possible if there is externally induced noise in the RTC_X1 and RTC_X2 pins.)
6. Ensure that the wiring of the crystal oscillator connection pins (RTC_X1 and RTC_X2) is routed as far away as possible from the power lines (except GND) and signal lines.
7. When not using a crystal oscillation circuit for RTC, fix the RTC_X1 pin (pull-up, pull-down, connect to power supply, or connect to ground) and leave the RTC_X2 pin open.

Figure 15.6 Example of Connecting Crystal Oscillator Circuit for RTC

15.5.5 Procedure for Setting the 30-Second Adjustment Function

Figure 15.7 shows the procedure for setting the 30-second adjustment function.

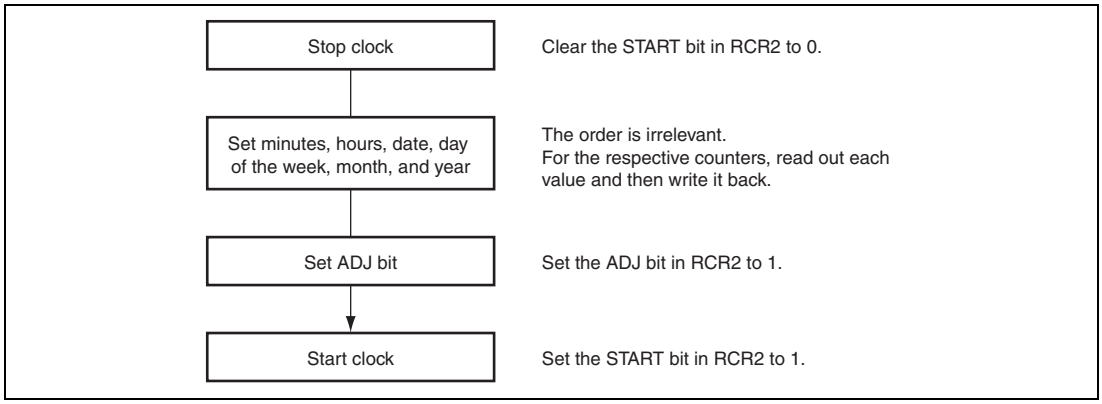


Figure 15.7 Procedure for Setting the 30-Second Adjustment Function

To use the 30-second adjustment function, the minutes, hours, date, day of the week, month, and year counters need to be written to. Thus, after clearing the START bit in RCR2 and reading out the minutes, hours, date, day of the week, month, and year counters and then writing the read values back, set the ADJ bit in RCR2 to 1. After the 30-second adjustment, set the START bit in RCR2 to 1 to start the clock operation.

Section 16 Serial Communication Interface with FIFO (SCIF)

This LSI has an eight-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.

16.1 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, break, receive-FIFO-data-full, 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 registers and the number of receive errors of the receive data in the receive FIFO register can be ascertained.
- A time-out error (DR) can be detected when receiving in asynchronous mode.

Figure 16.1 shows a block diagram of the SCIF.

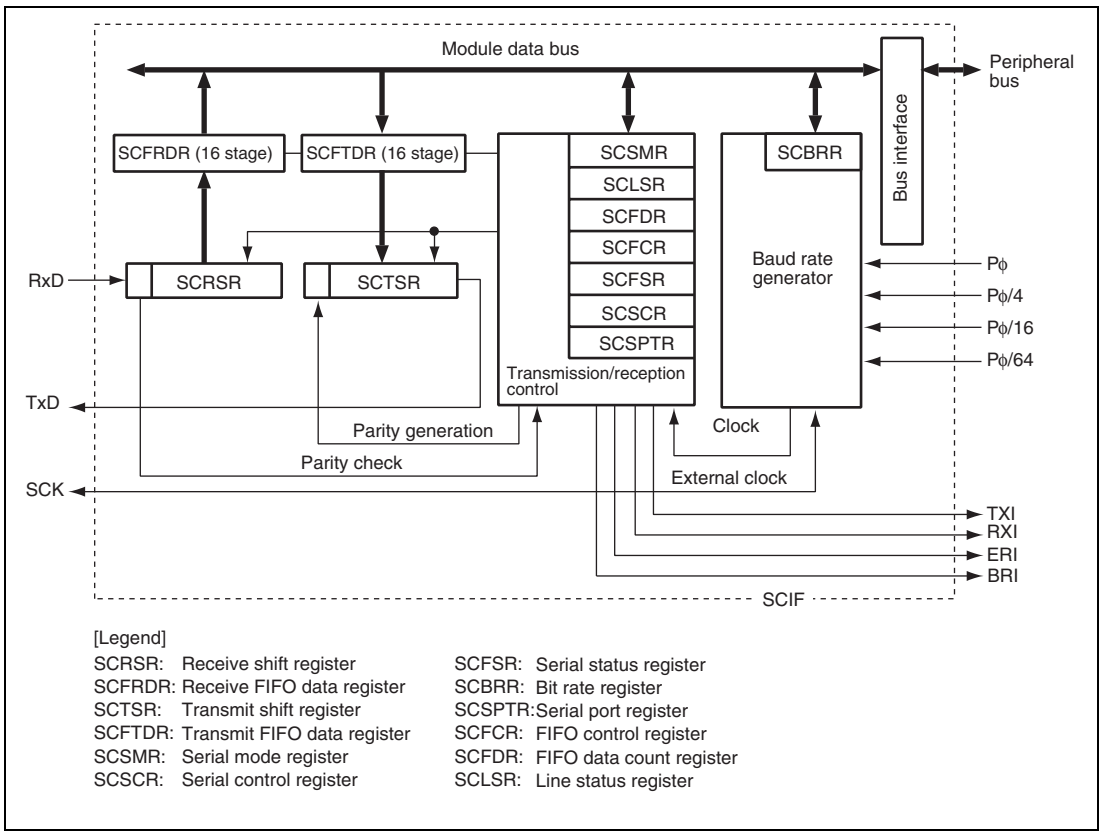


Figure 16.1 Block Diagram of SCIF

16.2 Input/Output Pins

Table 16.1 shows the pin configuration of the SCIF.

Table 16.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 7	Serial clock pins	SCK0 to SCK7	I/O	Clock I/O
	Receive data pins	RxD0 to RxD7	Input	Receive data input
	Transmit data pins	TxD0 to TxD7	Output	Transmit data output

16.3 Register Descriptions

The SCIF has the following registers.

Table 16.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Serial mode register_0	SCSMR_0	R/W	H'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

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
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
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
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

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
4	Serial mode register_4	SCSMR_4	R/W	H'0000	H'FFFEA000	16
	Bit rate register_4	SCBRR_4	R/W	H'FF	H'FFFEA004	8
	Serial control register_4	SCSCR_4	R/W	H'0000	H'FFFEA008	16
	Transmit FIFO data register_4	SCFTDR_4	W	Undefined	H'FFFEA00C	8
	Serial status register_4	SCFSR_4	R/(W)*1	H'0060	H'FFFEA010	16
	Receive FIFO data register_4	SCFRDR_4	R	Undefined	H'FFFEA014	8
	FIFO control register_4	SCFCR_4	R/W	H'0000	H'FFFEA018	16
	FIFO data count register_4	SCFDR_4	R	H'0000	H'FFFEA01C	16
	Serial port register_4	SCSPTR_4	R/W	H'0050	H'FFFEA020	16
	Line status register_4	SCLSR_4	R/(W)*2	H'0000	H'FFFEA024	16
5	Serial mode register_5	SCSMR_5	R/W	H'0000	H'FFFEA800	16
	Bit rate register_5	SCBRR_5	R/W	H'FF	H'FFFEA804	8
	Serial control register_5	SCSCR_5	R/W	H'0000	H'FFFEA808	16
	Transmit FIFO data register_5	SCFTDR_5	W	Undefined	H'FFFEA80C	8
	Serial status register_5	SCFSR_5	R/(W)*1	H'0060	H'FFFEA810	16
	Receive FIFO data register_5	SCFRDR_5	R	Undefined	H'FFFEA814	8
	FIFO control register_5	SCFCR_5	R/W	H'0000	H'FFFEA818	16
	FIFO data count register_5	SCFDR_5	R	H'0000	H'FFFEA81C	16
	Serial port register_5	SCSPTR_5	R/W	H'0050	H'FFFEA820	16
	Line status register_5	SCLSR_5	R/(W)*2	H'0000	H'FFFEA824	16
6	Serial mode register_6	SCSMR_6	R/W	H'0000	H'FFFEB000	16
	Bit rate register_6	SCBRR_6	R/W	H'FF	H'FFFEB004	8
	Serial control register_6	SCSCR_6	R/W	H'0000	H'FFFEB008	16
	Transmit FIFO data register_6	SCFTDR_6	W	Undefined	H'FFFEB00C	8
	Serial status register_6	SCFSR_6	R/(W)*1	H'0060	H'FFFEB010	16
	Receive FIFO data register_6	SCFRDR_6	R	Undefined	H'FFFEB014	8
	FIFO control register_6	SCFCR_6	R/W	H'0000	H'FFFEB018	16
	FIFO data count register_6	SCFDR_6	R	H'0000	H'FFFEB01C	16
	Serial port register_6	SCSPTR_6	R/W	H'0050	H'FFFEB020	16
	Line status register_6	SCLSR_6	R/(W)*2	H'0000	H'FFFEB024	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
7	Serial mode register_7	SCSMR_7	R/W	H'0000	H'FFFEB800	16
	Bit rate register_7	SCBRR_7	R/W	H'FF	H'FFFEB804	8
	Serial control register_7	SCSCR_7	R/W	H'0000	H'FFFEB808	16
	Transmit FIFO data register_7	SCFTDR_7	W	Undefined	H'FFFEB80C	8
	Serial status register_7	SCFSR_7	R/(W)*1	H'0060	H'FFFEB810	16
	Receive FIFO data register_7	SCFRDR_7	R	Undefined	H'FFFEB814	8
	FIFO control register_7	SCFCR_7	R/W	H'0000	H'FFFEB818	16
	FIFO data count register_7	SCFDR_7	R	H'0000	H'FFFEB81C	16
	Serial port register_7	SCSPTR_7	R/W	H'0050	H'FFFEB820	16
	Line status register_7	SCLSR_7	R/(W)*2	H'0000	H'FFFEB824	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.

16.3.1 Receive Shift Register (SCRSR)

SCRSR receives serial data. Data input at the RxD pin is loaded into SCRSR in the order received, LSB (bit 0) first, converting the data to parallel form. When one byte has been received, it is automatically transferred to 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:	—	—	—	—	—	—	—	—

16.3.2 Receive FIFO Data Register (SCFRDR)

SCFRDR is a 16-byte FIFO register that stores serial receive data. The SCIF completes the reception of one byte of serial data by moving the received data from the receive shift register (SCRSR) into SCFRDR for storage. Continuous reception is possible until 16 bytes are stored. The CPU can read but not write to SCFRDR. If data is read when there is no receive data in the SCFRDR, the value is undefined.

When SCFRDR is full of receive data, subsequent serial data is lost.

SCFRDR is initialized to an undefined value by a power-on reset or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R

16.3.3 Transmit Shift Register (SCTSR)

SCTSR transmits serial data. The 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:	—	—	—	—	—	—	—	—

16.3.4 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 or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
Initial value:	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W

16.3.5 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 or in deep standby mode.

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 the 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 16.3.8, Bit Rate Register (SCBRR). 00: Pϕ 01: Pϕ/4 10: Pϕ/16 11: Pϕ/64 Note: Pϕ: Peripheral clock

16.3.6 Serial Control Register (SCSCR)

SCSCR 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 or in deep standby mode.

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 interrupts (RXI) 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 SCIF 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 SCIF serial receiver. 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.
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	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 the combination of these bits, the SCK pin can be used for serial clock output or serial clock input. If serial clock output is set in clocked synchronous mode, the C/$\overline{A}$ bit in SCSMR is set to 1, and then these bits are set. • 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

16.3.7 Serial Status Register (SCFSR)

SCFSR is a 16-bit register. The upper 8 bits indicate the number of receives 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. SCFSR is initialized by a power-on reset or in deep standby mode.

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). After the ER bit in SCFSR is set, the value indicated by bits 15 to 12 represents the number of parity errors in SCFRDR. When parity errors have occurred in all 16-byte receive data in SCFRDR, PER3 to PER0 show 0.
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. After the ER bit in SCFSR is set, the value indicated by bits 11 to 8 represents the number of framing errors in SCFRDR. When framing errors have occurred in all 16-byte receive data in SCFRDR, FER3 to FER0 show 0.

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] 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: 1. 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 the DMAC is activated by the transmit FIFO data empty interrupt (TXI) and writes data exceeding the specified transmission trigger number to SCFTDR 1: The quantity of transmit data in SCFTDR is less than or equal to the specified transmission trigger number*[1] [Setting conditions] • TDFE is set to 1 by a power-on reset • TDFE is set to 1 when the quantity of transmit data in SCFTDR becomes less than or equal to the specified transmission trigger number as a result of transmission Note: 1. 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*[1] [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 1. 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 RTRG1 and RTRG0 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 DMAC read SCFRDR 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*[1] Note 1. 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 after the DMAC is activated by the receive FIFO data full interrupt (RXI) 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. *1 Note:1. 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.

16.3.8 Bit Rate Register (SCBRR)

SCBRR is an 8-bit register that, together with the baud rate generator clock source selected by the CKS1 and CKS0 bits in the serial mode register (SCSMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to SCBRR. SCBRR is initialized to H'FF by a power-on reset or in deep standby mode. Each channel has independent baud rate generator control, so different values can be set in eight 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:

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

- 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 16.3.)

Table 16.3 SCSMR Settings

		SCSMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

The bit rate error in asynchronous is given by the following formula:

$$\text{Error (\%)} = \left\{ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 16.4 lists examples of SCBRR settings in asynchronous mode, and table 16.5 lists examples of SCBRR settings in clocked synchronous mode.

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (1)

	Pϕ (MHz)											
	5			6			6.144			7.3728		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	88	–0.25	2	106	–0.44	2	108	0.08	2	130	–0.07
150	2	64	0.16	2	77	0.16	2	79	0.00	2	95	0.00
300	1	129	0.16	1	155	0.16	1	159	0.00	1	191	0.00
600	1	64	0.16	1	77	0.16	1	79	0.00	1	95	0.00
1200	0	129	0.16	0	155	0.16	0	159	0.00	0	191	0.00
2400	0	64	0.16	0	77	0.16	0	79	0.00	0	95	0.00
4800	0	32	–1.36	0	38	0.16	0	39	0.00	0	47	0.00
9600	0	15	1.73	0	19	–2.34	0	19	0.00	0	23	0.00
19200	0	7	1.73	0	9	–2.34	0	9	0.00	0	11	0.00
31250	0	4	0.00	0	5	0.00	0	5	2.40	0	6	5.33
38400	0	3	1.73	0	4	–2.34	0	4	0.00	0	5	0.00

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (2)

	Pϕ (MHz)											
	8			9.8304			10			12		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	0	6	–6.99	0	7	0.00	0	7	1.73	0	9	–2.34

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (3)

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

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (4)

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

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (5)

	Pϕ (MHz)														
	30			33			36			38			40		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	145	0.33	3	159	–0.12	3	168	–0.19	3	177	–0.25
150	3	97	–0.35	3	106	0.39	3	116	0.16	3	123	–0.24	3	129	0.16
300	2	194	0.16	2	214	–0.07	2	233	0.16	2	246	0.16	3	64	0.16
600	2	97	–0.35	2	106	0.39	2	116	0.16	2	123	–0.24	2	129	0.16
1200	1	194	0.16	1	214	–0.07	1	233	0.16	1	246	0.16	2	64	0.16
2400	1	97	–0.35	1	106	0.39	1	116	0.16	1	123	–0.24	1	129	0.16
4800	0	194	0.16	0	214	–0.07	0	233	0.16	0	246	0.16	1	64	0.16
9600	0	97	–0.35	0	106	0.39	0	116	0.16	0	123	–0.24	0	129	0.16
19200	0	48	–0.35	0	53	–0.54	0	58	–0.69	0	61	–0.24	0	64	0.16
31250	0	29	0.00	0	32	0.00	0	35	0.00	0	37	0.00	0	39	0.00
38400	0	23	1.73	0	26	–0.54	0	28	1.02	0	30	–0.24	0	32	–1.36

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

Table 16.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (1)

Bit Rate (bit/s)	Pϕ (MHz)									
	5		8		16		28.7		30	
	n	N	n	N	n	N	n	N	n	N
250	3	77	3	124	3	249				
500	3	38	2	249	3	124	3	223	3	233
1 k	2	77	2	124	2	249	3	111	3	116
2.5 k	1	124	1	199	2	99	2	178	2	187
5 k	0	249	1	99	1	199	2	89	2	93
10 k	0	124	0	199	1	99	1	178	1	187
25 k	0	49	0	79	0	159	1	71	1	74
50 k	0	24	0	39	0	79	0	143	0	149
100 k	—	—	0	19	0	39	0	71	0	74
250 k	0	4	0	7	0	15	—	—	0	29
500 k	—	—	0	3	0	7	—	—	0	14
1 M	—	—			0	3	—	—	—	—
2 M							—	—	—	—

Table 16.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (2)

Bit Rate (bit/s)	Pϕ (MHz)							
	33		36		38		40	
	n	N	n	N	n	N	n	N
250								
500	3	255	—	—				
1 k	3	128	3	140	3	147	3	155
2.5 k	2	205	2	224	2	237	2	249
5 k	2	102	2	112	2	118	2	124
10 k	1	205	1	224	1	237	1	249
25 k	1	82	1	89	1	94	1	99
50 k	0	164	0	179	0	189	0	199
100 k	0	82	0	89	0	94	0	99
250 k	0	32	0	35	0	37	0	39
500 k	—	—	0	17	0	18	0	19
1 M	—	—	0	8	—	—	0	9
2 M	—	—	—	—	—	—	0	4

[Legend]

Blank: No setting possible, or it is not possible to satisfy the electrical characteristics of the MCU regardless of the communication partner device.

—: Setting possible, but error occurs

Table 16.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 16.7 and 16.8 list the maximum rates when the external clock input is used (when tscyc = 12 tpcyc*).

Note: * Make sure that the electrical characteristics of this MCU and that of a connected MCU are satisfied.

Table 16.6 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)

		Settings	
Pϕ (MHz)	**Maximum Bit Rate (bits/s)**	**n**	**N**
5	156250	0	0
8	250000	0	0
9.8304	307200	0	0
12	375000	0	0
14.7456	460800	0	0
16	500000	0	0
19.6608	614400	0	0
20	625000	0	0
24	750000	0	0
24.576	768000	0	0
28.7	896875	0	0
30	937500	0	0
33	1031250	0	0
36	1125000	0	0
38	1187500	0	0
40	1250000	0	0

Table 16.7 Maximum Bit Rates with External Clock Input (Asynchronous Mode)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	1.2500	78125
8	2.0000	125000
9.8304	2.4576	153600
12	3.0000	187500
14.7456	3.6864	230400
16	4.0000	250000
19.6608	4.9152	307200
20	5.0000	312500
24	6.0000	375000
24.576	6.1440	384000
28.7	7.1750	448436
30	7.5000	468750
33	8.2500	515625
36	9.0000	562500
38	9.5000	593750
40	10.0000	625000

Table 16.8 Maximum Bit Rates with External Clock Input (Clocked Synchronous Mode, $t_{Scyc} = 12\ t_{pcyc}$)

Pϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
5	0.4166	416666.6
8	0.6666	666666.6
16	1.3333	1333333.3
24	2.0000	2000000.0
28.7	2.3916	2391666.6
30	2.5000	2500000.0
33	2.7500	2750000.0
36	3.0000	3000000.0
38	3.1666	3166666.6
40	3.3333	3333333.3

16.3.9 FIFO Control Register (SCFCR)

SCFCR resets the quantity of data in the transmit and receive data FIFO 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 or in deep standby mode.

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 a number other than 1 is set, 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 enables loop-back testing. 0: Loop back test disabled 1: Loop back test enabled

16.3.10 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 or in deep standby mode.

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.

16.3.11 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 or in deep standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	SCKIO	SCKDT	SPB2 IO	SPB2 DT
Initial value:	0	0	0	0	0	0	0	0	0	1	0	1	0	—	0	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
5	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	SCKIO	0	R/W	SCK Port Input/Output Indicates input or output of the serial port SCK pin. When the SCK pin is actually used as a port outputting the SCKDT bit value, the CKE[1:0] bits in SCSCR should be cleared to 0. 0: SCKDT bit value not output to SCK pin 1: SCKDT bit value output to SCK pin

Bit	Bit Name	Initial Value	R/W	Description
2	SCKDT	Undefined	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
1	SPB2IO	0	R/W	Serial Port Break Input/Output Indicates input or output of the serial port TxD pin. When the TxD pin is actually used as a port outputting the SPB2DT bit value, the TE bit in SCSCR should be cleared to 0. 0: SPB2DT bit value not output to TxD pin 1: SPB2DT bit value output to TxD pin
0	SPB2DT	Undefined	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

16.3.12 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 or in deep standby mode.

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.

16.4 Operation

16.4.1 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 16.9. 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 16.10.

(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 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 on-chip baud rate generator, and outputs a serial clock signal to external devices.
  - When an external clock is selected, the SCIF operates on the input serial clock. The on-chip baud rate generator is not used.

Table 16.9 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 16.10 SCSMR and SCSCR Settings and SCIF Clock Source Selection

SCSMR	SCSCR Settings			SCIF Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Asynchronous	Internal	SCIF does not use the SCK pin
		1			Outputs a clock with a frequency 16 times the bit rate
	1	0		External	Inputs a clock with frequency 16 times the bit rate
		1		Setting prohibited	
1	0	x	Clocked synchronous	Internal	Outputs the serial clock
	1	0		External	Inputs the serial clock
		1		Setting prohibited	

[Legend]

x: Don't care

16.4.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the 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 16.2 shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The 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.

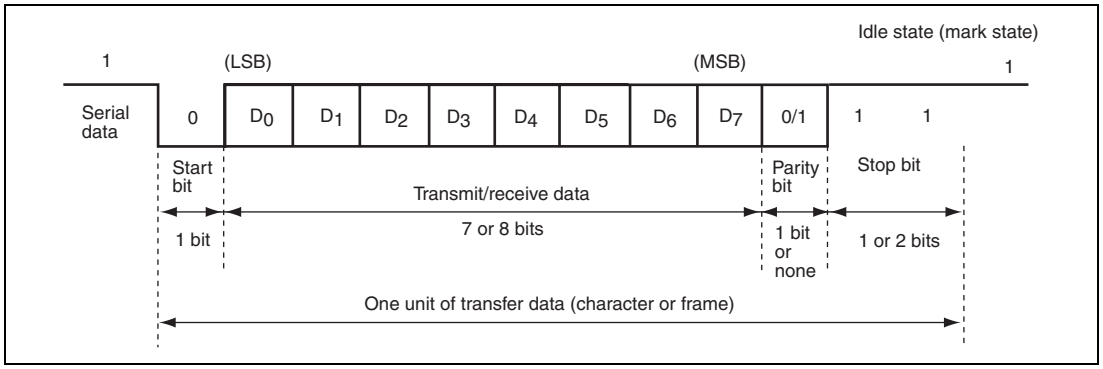


Figure 16.2 Example of Data Format in Asynchronous Communication (8-Bit Data with Parity and Two Stop Bits)

(1) Transmit/Receive Formats

Table 16.11 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 16.11 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 CKE1 and CKE0 in the serial control register (SCSCR). For clock source selection, refer to table 16.10.

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCIF operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16 times the desired bit rate.

(3) Transmitting and Receiving Data

- **SCIF Initialization (Asynchronous Mode)**

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCSCR), then initialize the SCIF as follows.

When changing the operation 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 16.3 shows a sample flowchart for initializing the SCIF.

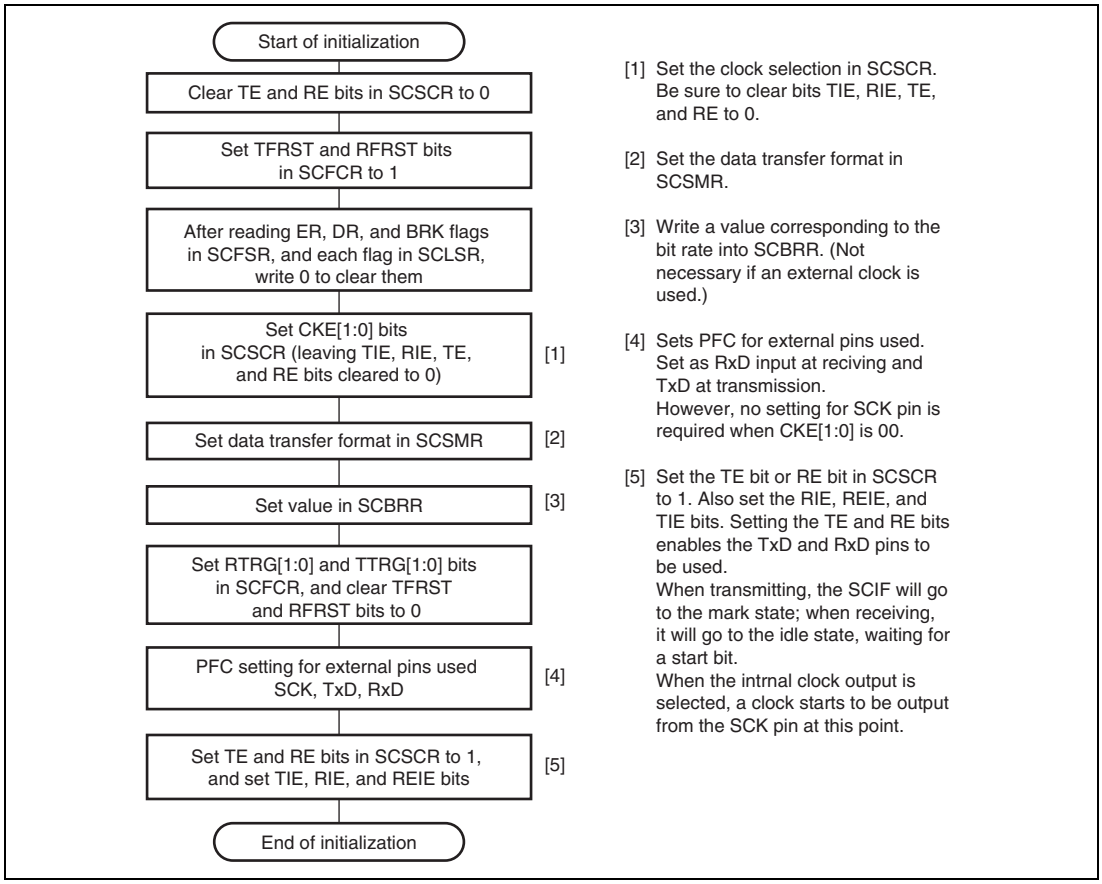


Figure 16.3 Sample Flowchart for SCIF Initialization

- **Transmitting Serial Data (Asynchronous Mode)**

Figure 16.4 shows a sample flowchart for serial transmission.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

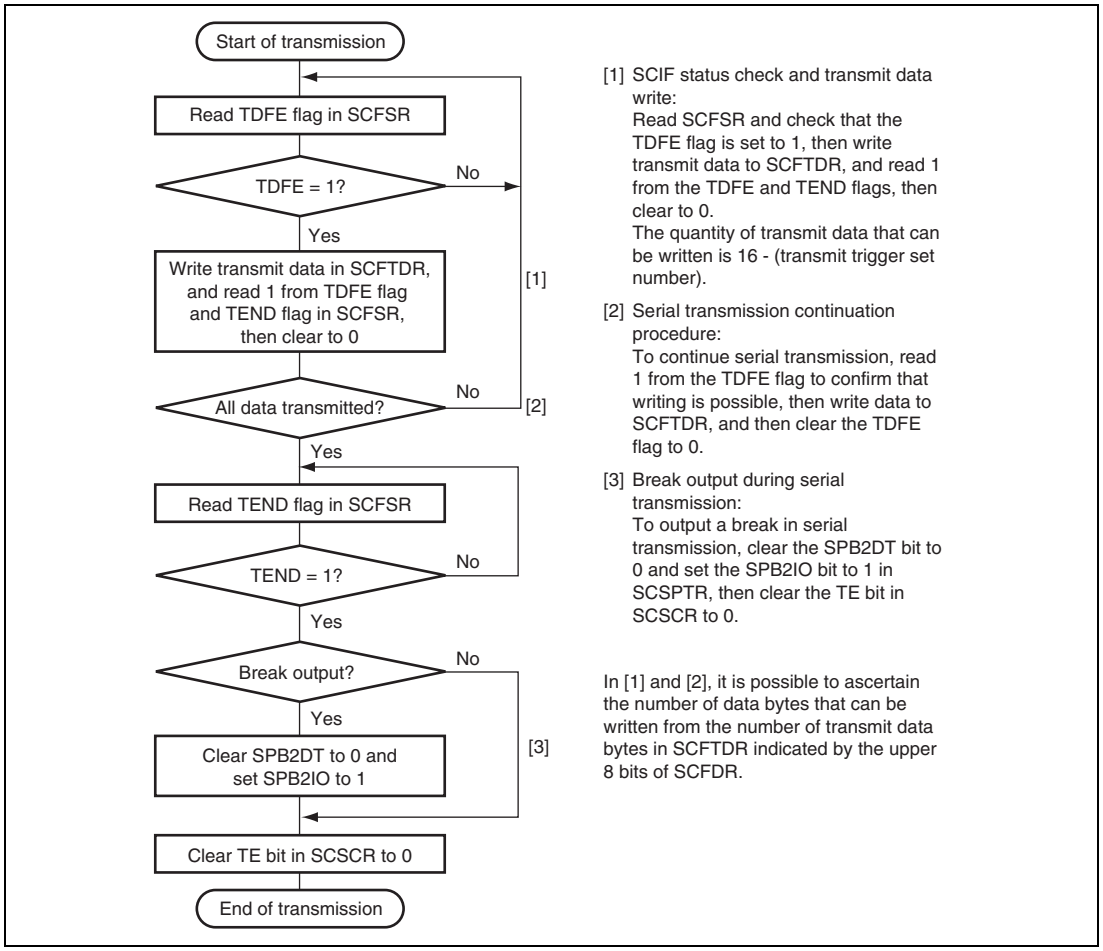


Figure 16.4 Sample Flowchart for Transmitting Serial Data

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 16.5 shows an example of the operation for transmission.

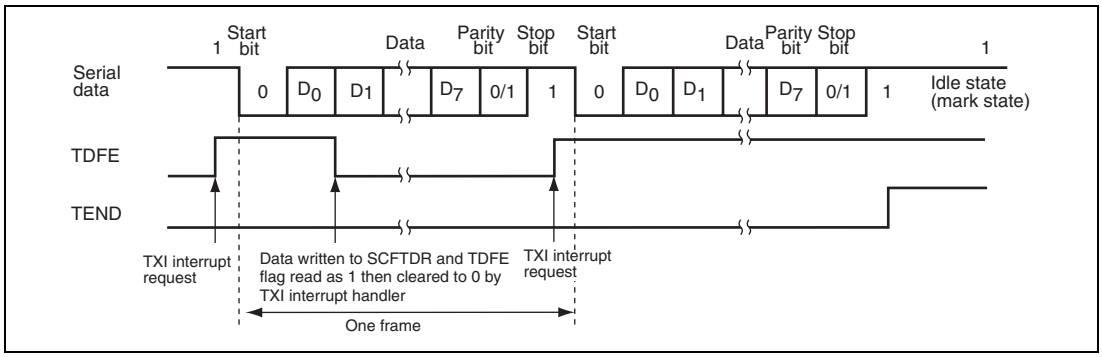


Figure 16.5 Example of Transmit Operation (8-Bit Data, Parity, 1 Stop Bit)

- **Receiving Serial Data (Asynchronous Mode)**

Figures 16.6 and 16.7 show sample flowcharts for serial reception.

Use the following procedure for serial data reception after enabling the SCIF for reception.

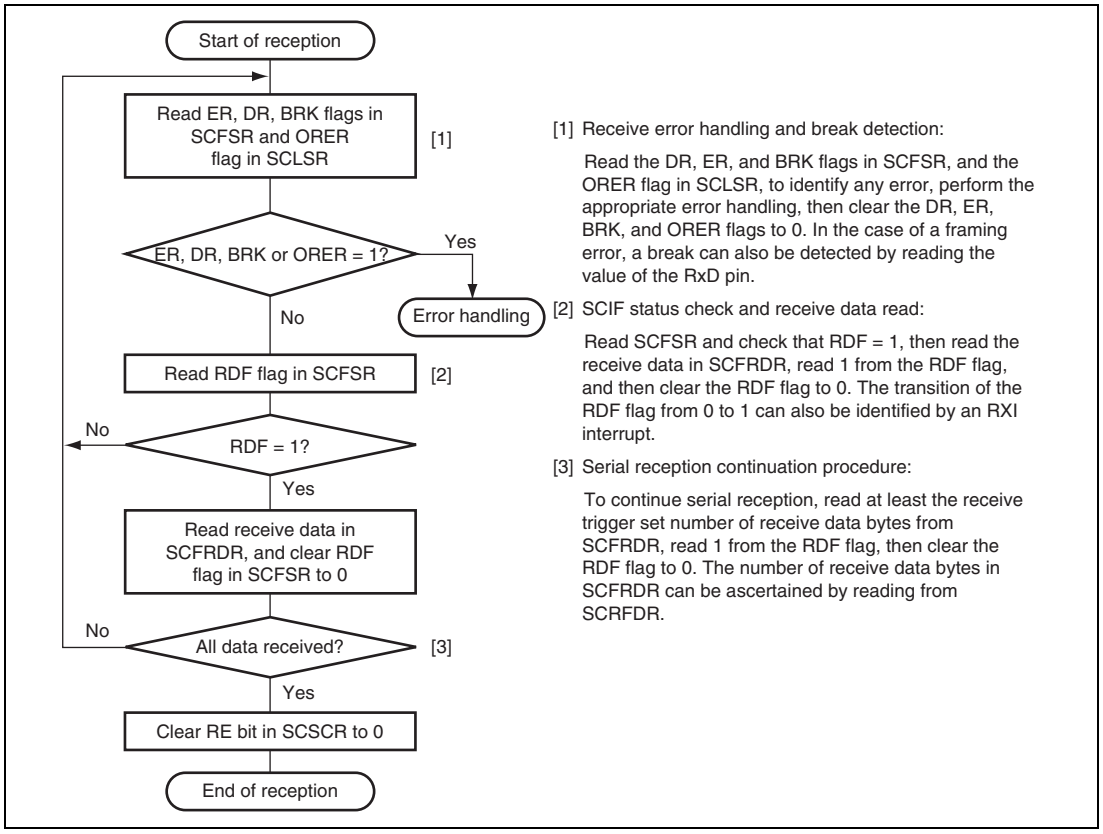
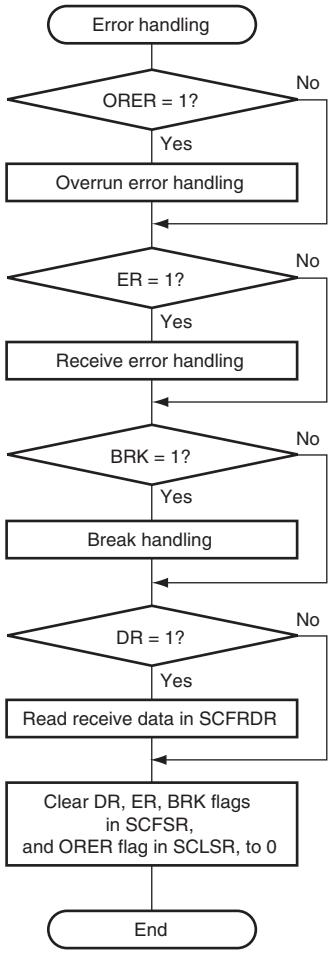


Figure 16.6 Sample Flowchart for Receiving Serial Data



- 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 16.7 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 16.8 shows an example of the operation for reception.

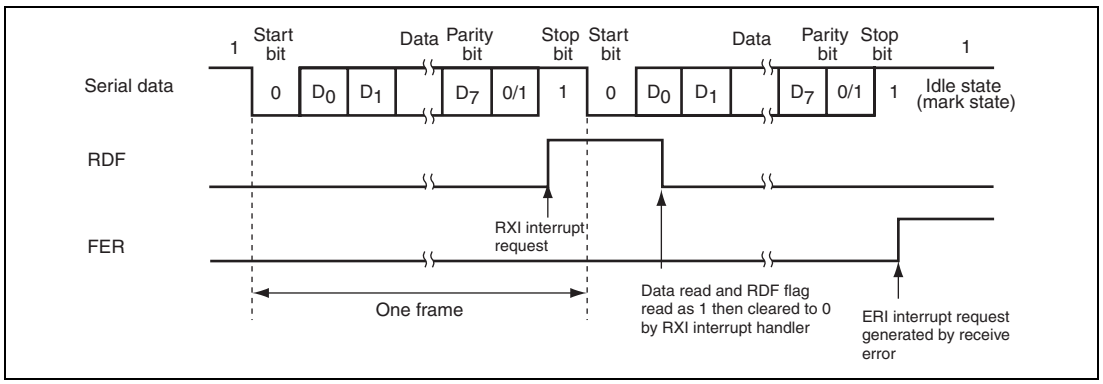


Figure 16.8 Example of SCIF Receive Operation (8-Bit Data, Parity, 1 Stop Bit)

16.4.3 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 16.9 shows the general format in clocked synchronous serial communication.

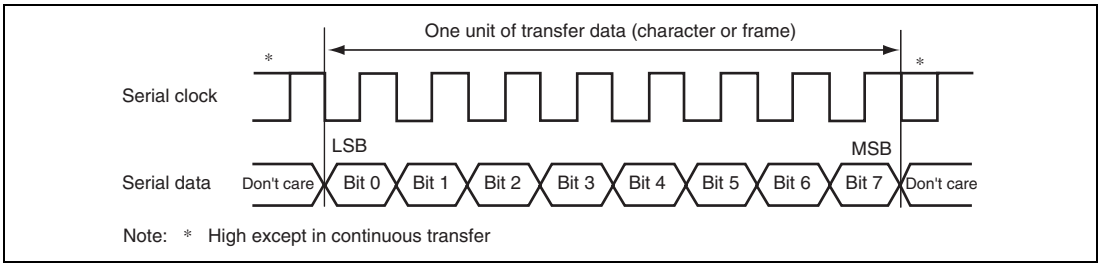


Figure 16.9 Data Format in Clocked Synchronous Communication

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/$\overline{A}$ 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 16.10 shows a sample flowchart for initializing the SCIF.

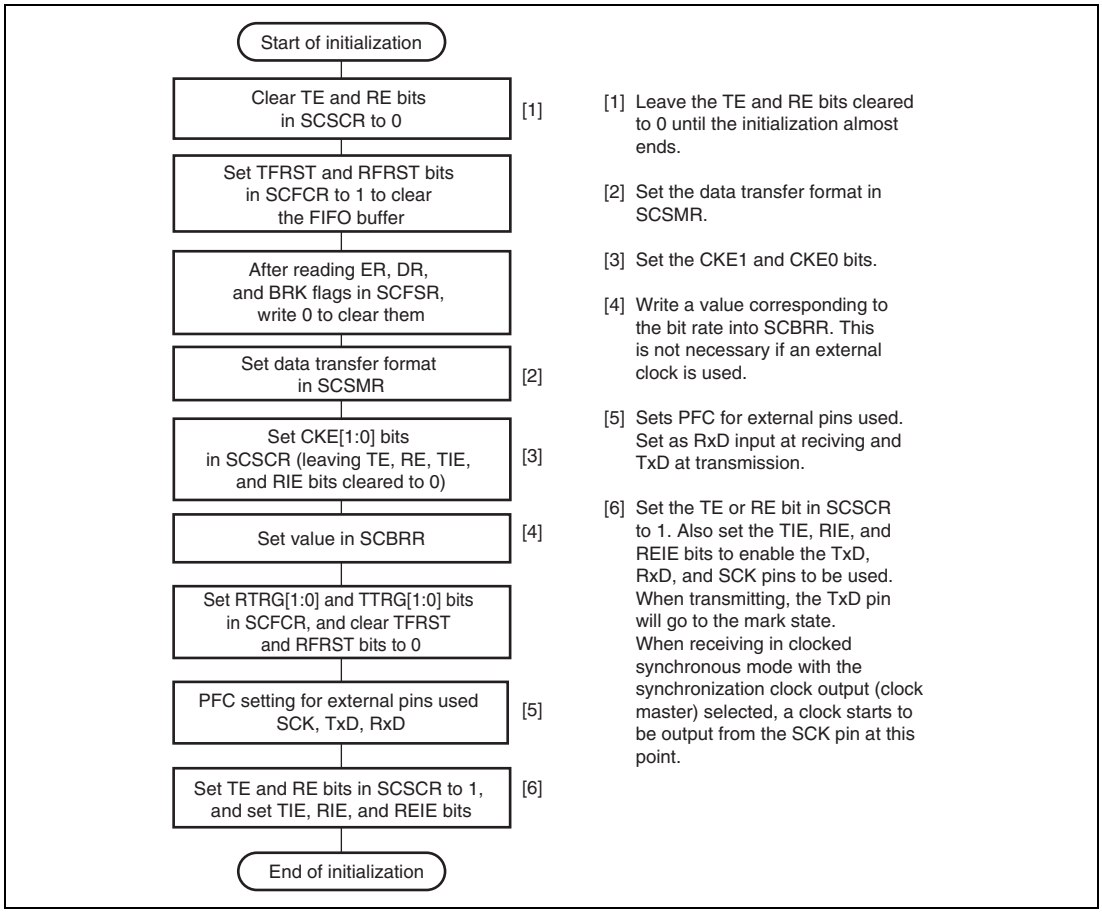


Figure 16.10 Sample Flowchart for SCIF Initialization

- **Transmitting Serial Data (Clocked Synchronous Mode)**

Figure 16.11 shows a sample flowchart for transmitting serial data.

Use the following procedure for serial data transmission after enabling the SCIF for transmission.

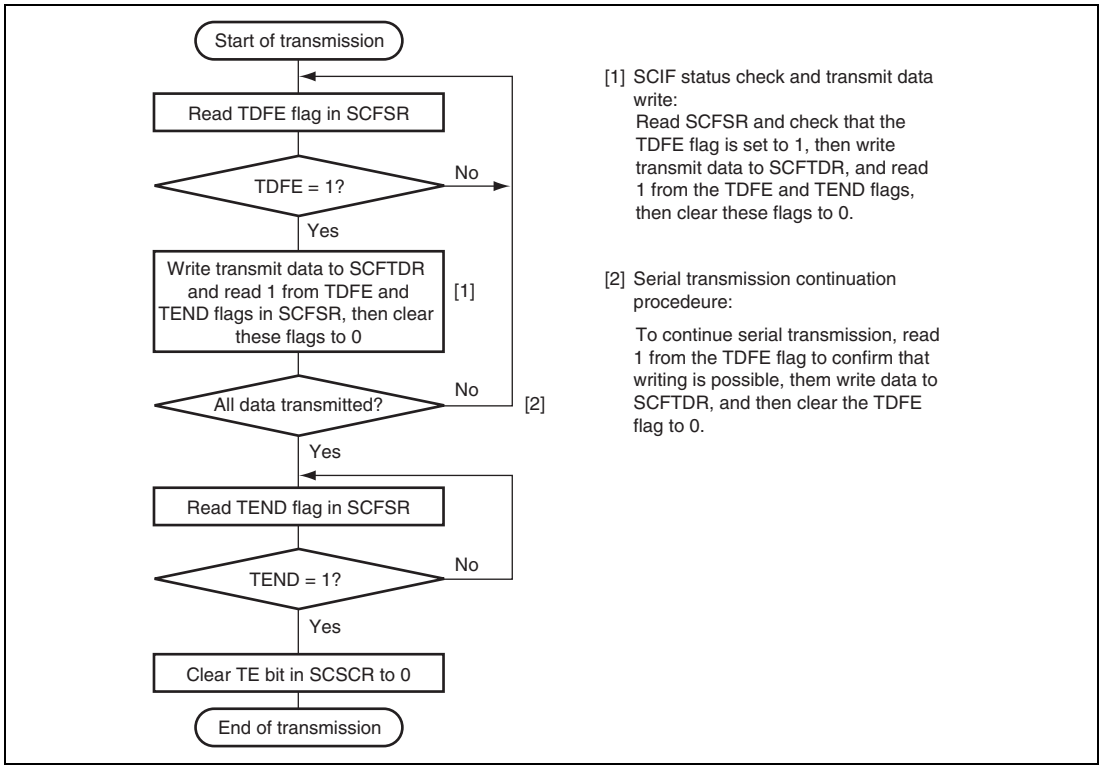


Figure 16.11 Sample Flowchart for Transmitting Serial Data

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 16.12 shows an example of SCIF transmit operation.

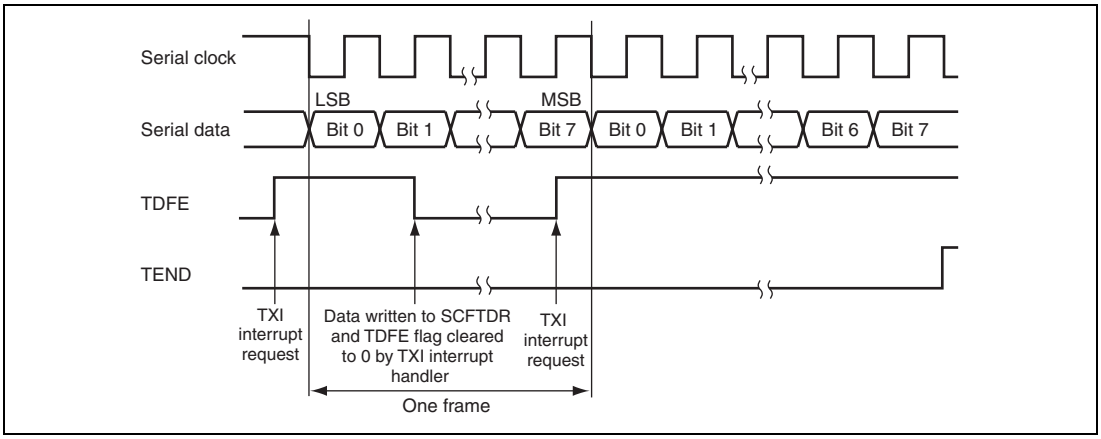


Figure 16.12 Example of SCIF Transmit Operation

- **Receiving Serial Data (Clocked Synchronous Mode)**

Figures 16.13 and 16.14 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.

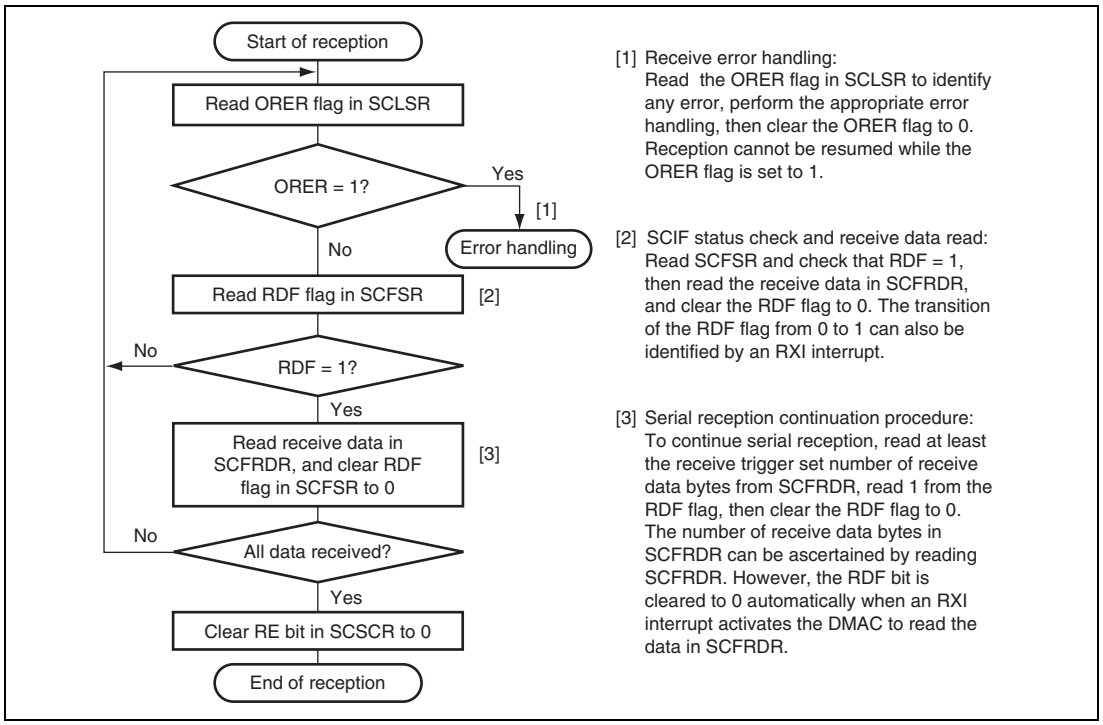


Figure 16.13 Sample Flowchart for Receiving Serial Data (1)

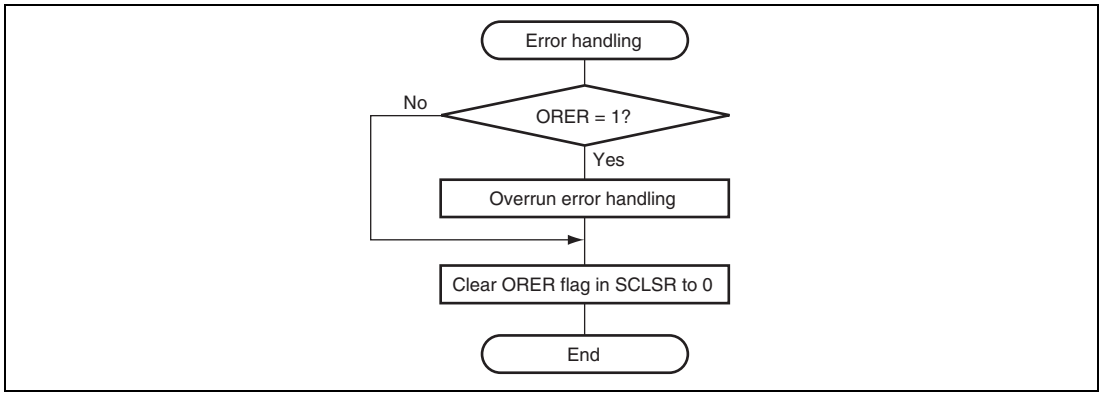


Figure 16.14 Sample Flowchart for Receiving Serial Data (2)

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-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 16.15 shows an example of SCIF receive operation.

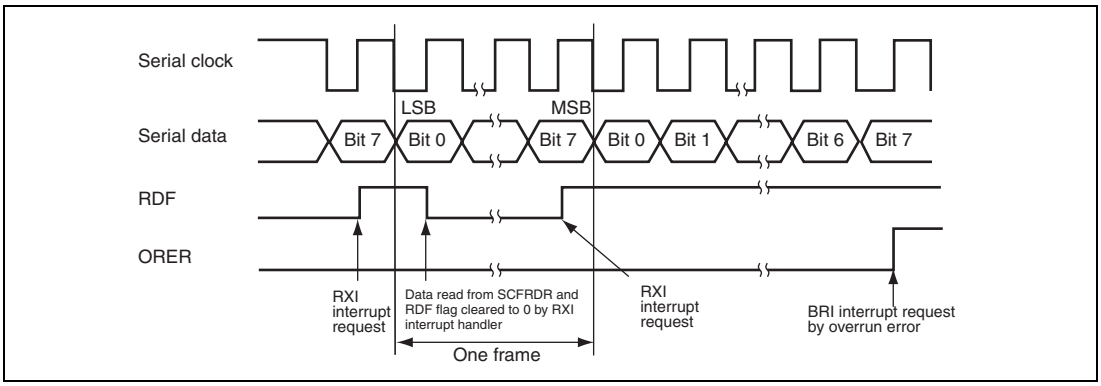


Figure 16.15 Example of SCIF Receive Operation

- **Transmitting and Receiving Serial Data Simultaneously (Clocked Synchronous Mode)**

Figure 16.16 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.

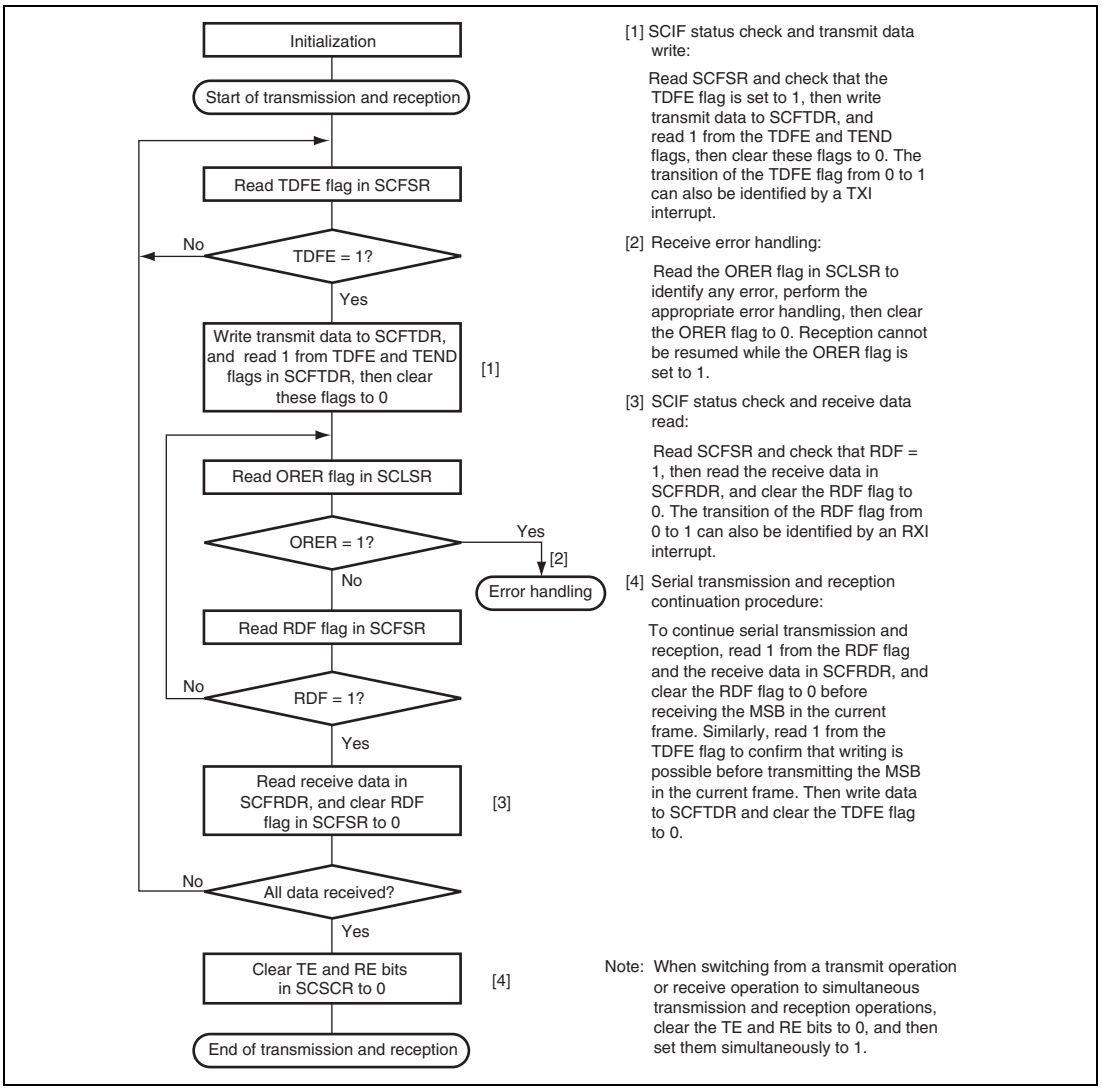


Figure 16.16 Sample Flowchart for Transmitting/Receiving Serial Data

16.5 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 16.12 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 RDF 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 or a BRI 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 16.12 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

16.6 Usage Notes

Note the following when using the SCIF.

16.6.1 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 TTRG1 and TTRG0 in the FIFO control register (SCFCR). After TDFE 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 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).

16.6.2 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 RTRG1 and RTRG0 in the FIFO control register (SCFCR). After RDF 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. RDF 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).

16.6.3 Restriction on DMAC Usage

1. When the DMAC writes data to SCFTDR with 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.
2. When one channel is used in full duplex communication with the DMAC used for transmission and the CPU used for reception, if the receive data are read from the receive FIFO data register (SCFRDR) after the RDF or DR flag in the serial status register (SCFSR) has been set, the RDF or DR flag may be cleared.
3. When one channel is used in full duplex communication with the DMAC used for reception and the CPU used for transmission, if the transmit data is written to the transmit FIFO data register (SCFTDR) after the TDFE or TEND flag in the serial status register (SCFSR) has been set, the TDFE or TEND flags may be cleared.

16.6.4 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.

16.6.5 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.

16.6.6 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 16.17.

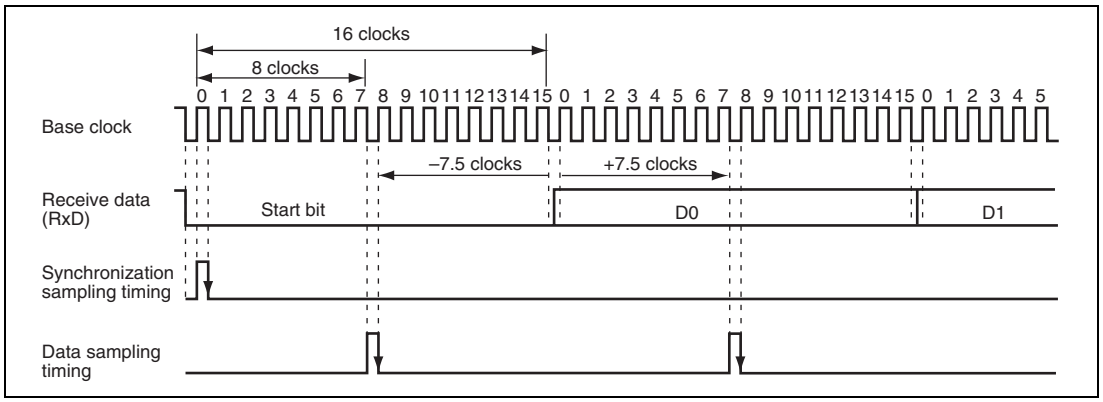


Figure 16.17 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

Equation 1:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ \%$$

Where: M: Receive margin (%)
N: Ratio of clock frequency to bit rate (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute deviation of clock frequency

From equation 1, if F = 0 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

Equation 2:

When D = 0.5 and F = 0:

$M = (0.5 - 1/(2 \times 16)) \times 100\%$

$= 46.875\%$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

Section 17 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.

17.1 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, SCL0 to SCL2 and SDA0 to SDA2, 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 17.1 shows a block diagram of the I²C bus interface 3.

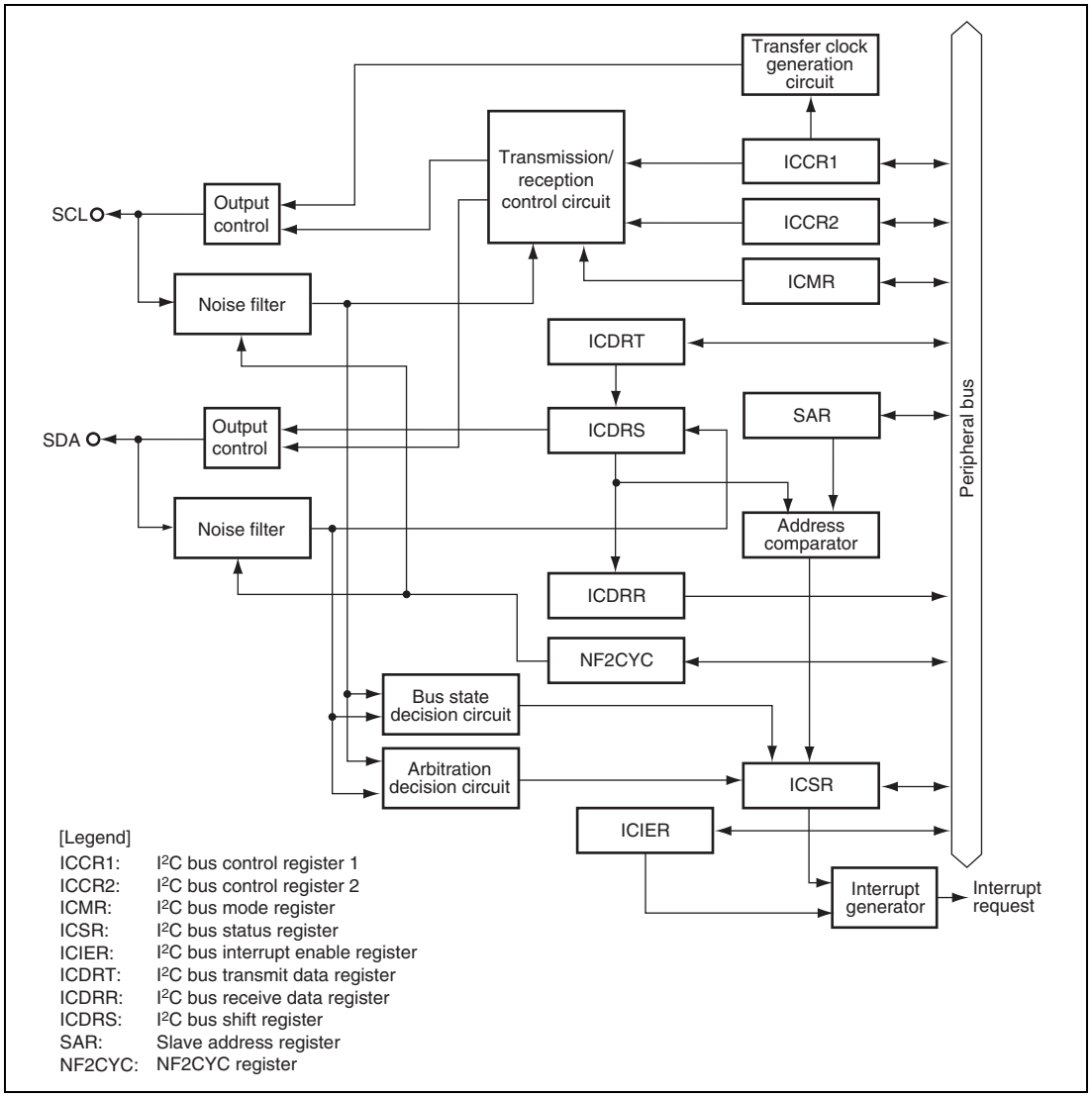


Figure 17.1 Block Diagram of I²C Bus Interface 3

17.2 Input/Output Pins

Table 17.1 shows the pin configuration of the I²C bus interface 3. Specifications for the voltage applied to I/O pins for the I²C bus interface are different from others because of the pin configuration difference. For details, see section 29, Electrical Characteristics.

Table 17.1 Pin Configuration

Channel	Pin Name	Symbol	I/O	Function
0 to 2	Serial clock	SCL0 to SCL2	I/O	I²C serial clock input/output
	Serial data	SDA0 to SDA2	I/O	I²C serial data input/output

Figure 17.2 shows an example of I/O pin connections to external circuits.

Specifications for the voltage applied to I/O pins for the I²C bus interface are different from others because of the pin configuration difference. For details, see section 29, Electrical Characteristics.

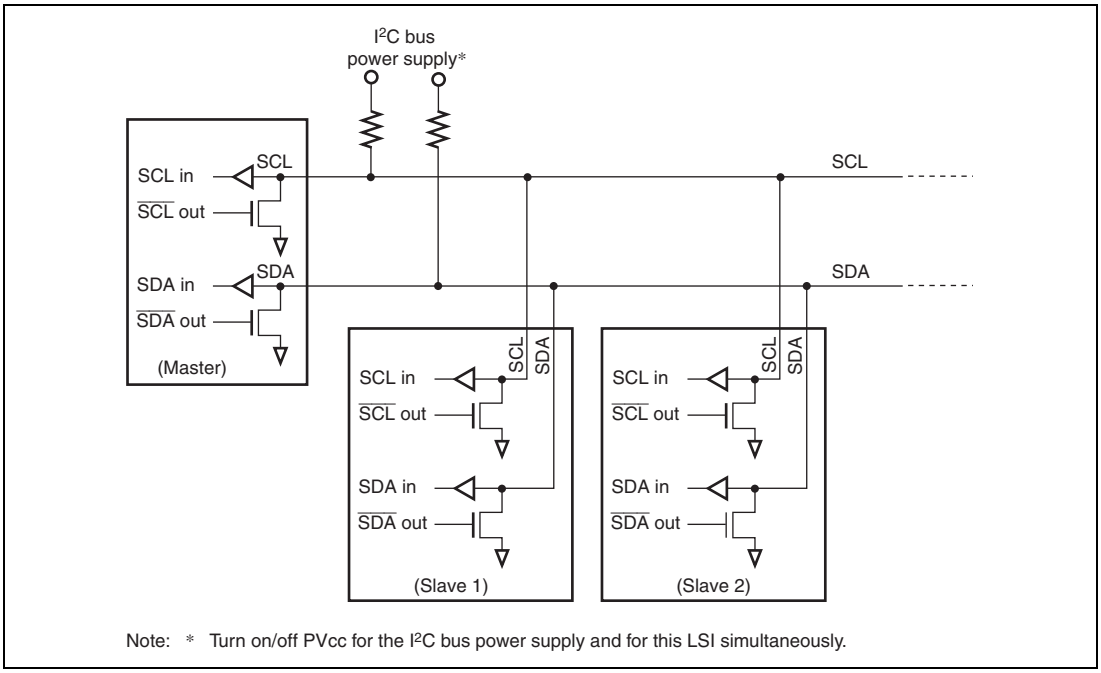


Figure 17.2 External Circuit Connections of I/O Pins

17.3 Register Descriptions

The I²C bus interface 3 has the following registers.

Table 17.2 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	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'02	H'FFFEE008	8
1	I²C bus control register 1	ICCR1	R/W	H'00	H'FFFEE080	8
	I²C bus control register 2	ICCR2	R/W	H'7D	H'FFFEE081	8
	I²C bus mode register	ICMR	R/W	H'38	H'FFFEE082	8
	I²C bus interrupt enable register	ICIER	R/W	H'00	H'FFFEE083	8
	I²C bus status register	ICSR	R/W	H'00	H'FFFEE084	8
	Slave address register	SAR	R/W	H'00	H'FFFEE085	8
	I²C bus transmit data register	ICDRT	R/W	H'FF	H'FFFEE086	8
	I²C bus receive data register	ICDRR	R/W	H'FF	H'FFFEE087	8
	NF2CYC register	NF2CYC	R/W	H'02	H'FFFEE088	8
2	I²C bus control register 1	ICCR1	R/W	H'00	H'FFFEE100	8
	I²C bus control register 2	ICCR2	R/W	H'7D	H'FFFEE101	8
	I²C bus mode register	ICMR	R/W	H'38	H'FFFEE102	8
	I²C bus interrupt enable register	ICIER	R/W	H'00	H'FFFEE103	8
	I²C bus status register	ICSR	R/W	H'00	H'FFFEE104	8
	Slave address register	SAR	R/W	H'00	H'FFFEE105	8
	I²C bus transmit data register	ICDRT	R/W	H'FF	H'FFFEE106	8
	I²C bus receive data register	ICDRR	R/W	H'FF	H'FFFEE107	8
	NF2CYC register	NF2CYC	R/W	H'02	H'FFFEE108	8

17.3.1 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 or deep standby mode.

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: SCL and SDA output is disabled. (Input to SCL and SDA is enabled.) 1: This bit is enabled for transfer operations.
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception

Bit	Bit Name	Initial Value	R/W	Description
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select In master mode with the 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 17.3) in master mode.

Table 17.3 Transfer Rate

Bit 3	Bit 2	Bit 1	Bit 0		Transfer Rate					
CKS3	CKS2	CKS1	CKS0	Clock	Pϕ = 16.7 MHz	Pϕ = 20.0 MHz	Pϕ = 25.0 MHz	Pϕ = 30.0 MHz	Pϕ = 33.3 MHz	Pϕ = 40 MHz
0	0	0	0	Pϕ/28	595 kHz	714 kHz	893 kHz	1071 kHz	1189 kHz	1430 kHz
			1	Pϕ/40	417 kHz	500 kHz	625 kHz	750 kHz	833 kHz	1000 kHz
		1	0	Pϕ/48	347 kHz	417 kHz	521 kHz	625 kHz	694 kHz	833 kHz
			1	Pϕ/64	260 kHz	313 kHz	391 kHz	469 kHz	520 kHz	625 kHz
	1	0	0	Pϕ/80	208 kHz	250 kHz	313 kHz	375 kHz	416 kHz	500 kHz
			1	Pϕ/100	167 kHz	200 kHz	250 kHz	300 kHz	333 kHz	400 kHz
		1	0	Pϕ/112	149 kHz	179 kHz	223 kHz	268 kHz	297 kHz	357 kHz
			1	Pϕ/128	130 kHz	156 kHz	195 kHz	234 kHz	260 kHz	313 kHz
1	0	0	0	Pϕ/112	149 kHz	179 kHz	223 kHz	268 kHz	297 kHz	357 kHz
			1	Pϕ/160	104 kHz	125 kHz	156 kHz	188 kHz	208 kHz	250 kHz
		1	0	Pϕ/192	86.8 kHz	104 kHz	130 kHz	156 kHz	173 kHz	208 kHz
			1	Pϕ/256	65.1 kHz	78.1 kHz	97.7 kHz	117 kHz	130 kHz	156 kHz
	1	0	0	Pϕ/320	52.1 kHz	62.5 kHz	78.1 kHz	93.8 kHz	104 kHz	125 kHz
			1	Pϕ/400	41.7 kHz	50.0 kHz	62.5 kHz	75.0 kHz	83.3 kHz	100 kHz
		1	0	Pϕ/448	37.2 kHz	44.6 kHz	55.8 kHz	67.0 kHz	74.3 kHz	89.3 kHz
			1	Pϕ/512	32.6 kHz	39.1 kHz	48.8 kHz	58.6 kHz	65.0 kHz	78.1 kHz

Note: The settings should satisfy external specifications.

17.3.2 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 or deep standby mode.

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 bits BC[2:0] in ICMR and internal circuits. If this bit is set to 1 when hang-up occurs because of communication failure during I²C bus operation, bits BC[2:0] in ICMR and internal circuits can be reset.
0	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

17.3.3 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 or deep standby mode. 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	WAIT	—	—	BCWP	BC[2:0]		
Initial value:	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MLS	0	R/W	MSB-First/LSB-First Select 0: MSB-first 1: LSB-first Set this bit to 0 when the I²C bus format is used.
6	WAIT	0	R/W	Wait Insertion In master mode with the I²C bus format, this bit selects whether to insert a wait after data transfer except the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode with the I²C bus format or with the clocked synchronous serial format.
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. The bit value returns to B'000 automatically at the end of a data transfer including the acknowledge bit. And the value becomes B'111 automatically after the stop condition detection. These bits are cleared by a power-on reset, in deep standby mode, software standby mode, or 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

17.3.4 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 or deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1 or 0, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) are enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection, arbitration lost and overrun error interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: 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.

17.3.5 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 or deep standby mode.

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 Register 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 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.

Bit	Bit Name	Initial Value	R/W	Description
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

17.3.6 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 or deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	SVA[6:0]							FS
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 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

17.3.7 I²C Bus Transmit Data Register (ICDRT)

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the empty 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.

ICDRT is initialized to H'FF by a power-on reset or deep standby mode.

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

17.3.8 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 or deep standby mode.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R

17.3.9 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:	—	—	—	—	—	—	—	—

17.3.10 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 17.4.7, Noise Filter.

NF2CYC is initialized to H'02 by a power-on reset or in deep standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	NF2CYC
Initial value:	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	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	—	1	R	Reserved This bit is always read as 0. The write value should always be 1.
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

17.4 Operation

The I²C bus interface 3 can communicate either in I²C bus mode or clocked synchronous serial mode by setting FS in SAR.

17.4.1 I²C Bus Format

Figure 17.3 shows the I²C bus formats. Figure 17.4 shows the I²C bus timing. The first frame following a start condition always consists of eight bits.

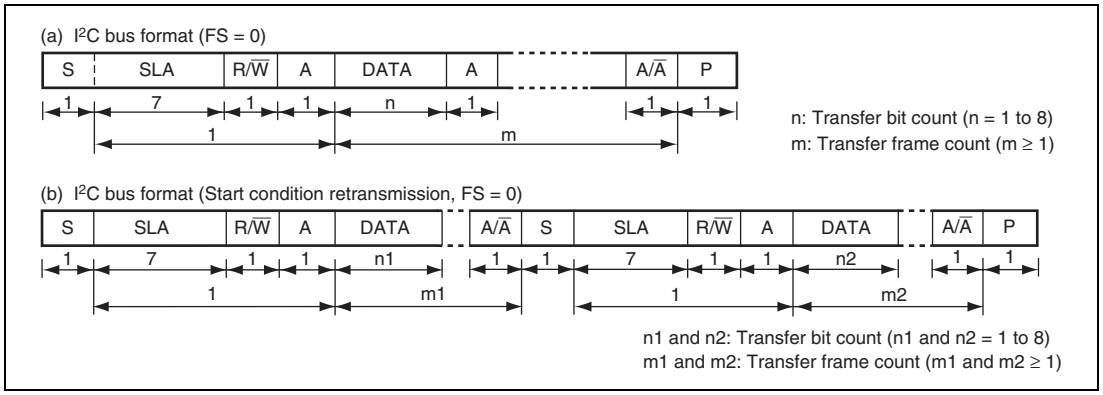


Figure 17.3 I²C Bus Formats

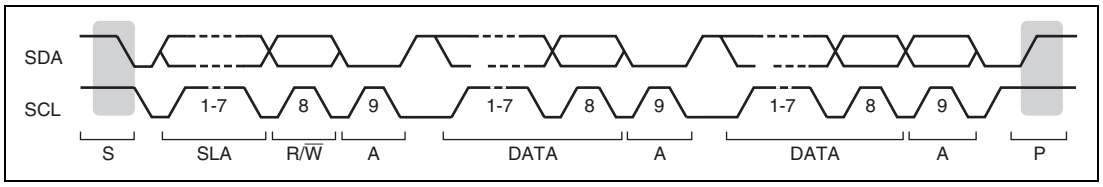


Figure 17.4 I²C Bus Timing

[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.

17.4.2 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 17.5 and 17.6. The transmission procedure and operations in master transmit mode are described below.

1. Set the ICE bit in ICCR1 to 1. Set the WAIT bit in ICMR and bits CKS[3:0] in ICCR1. (Initial setting)
2. Read the BBSY flag in ICCR2 to confirm that the bus is released. Set the MST and TRS bits in ICCR1 to select master transmit mode. Then, write 1 to BBSY and 0 to SCP. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/$\overline{W}$) to ICDRT. At this time, TDRE is automatically cleared to 0, and data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the receive device while ACKE in ICIER is 1. Then, issue the stop condition to clear TEND or NACKF.
7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.

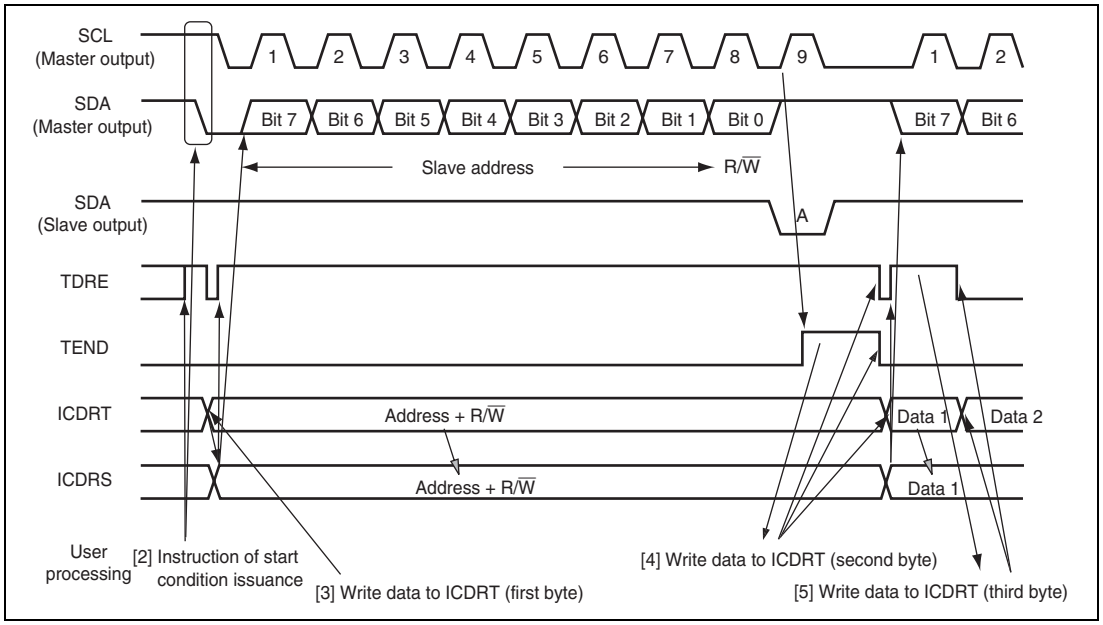


Figure 17.5 Master Transmit Mode Operation Timing (1)

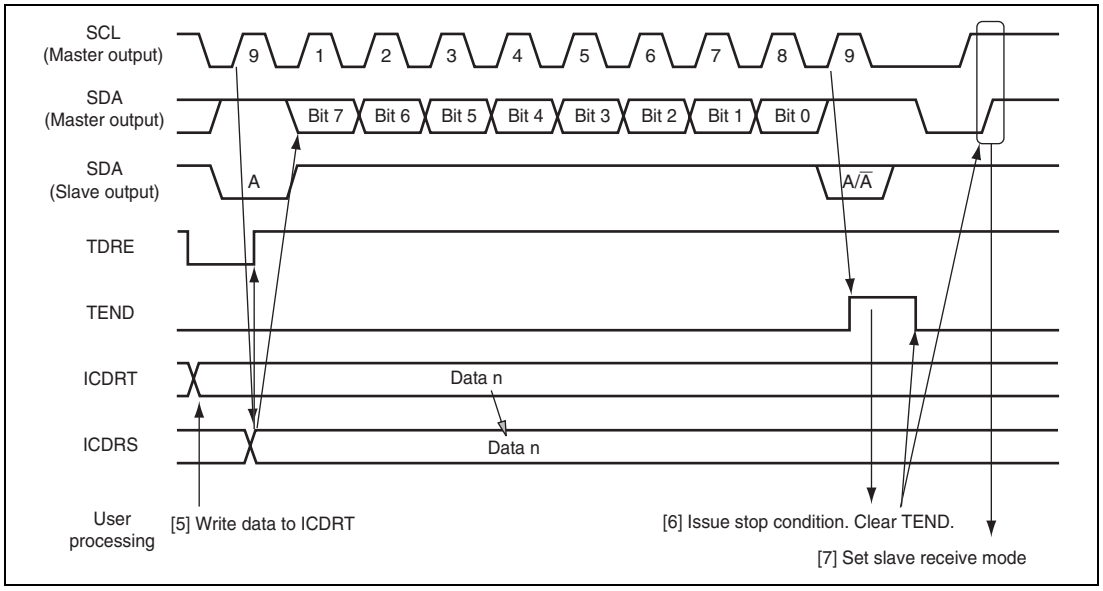


Figure 17.6 Master Transmit Mode Operation Timing (2)

17.4.3 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 17.7 and 17.8. 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 the slave receive mode.

Note: * If only one byte is received, read ICDRR (dummy-read) after the RCVD bit in ICCR1 is set.

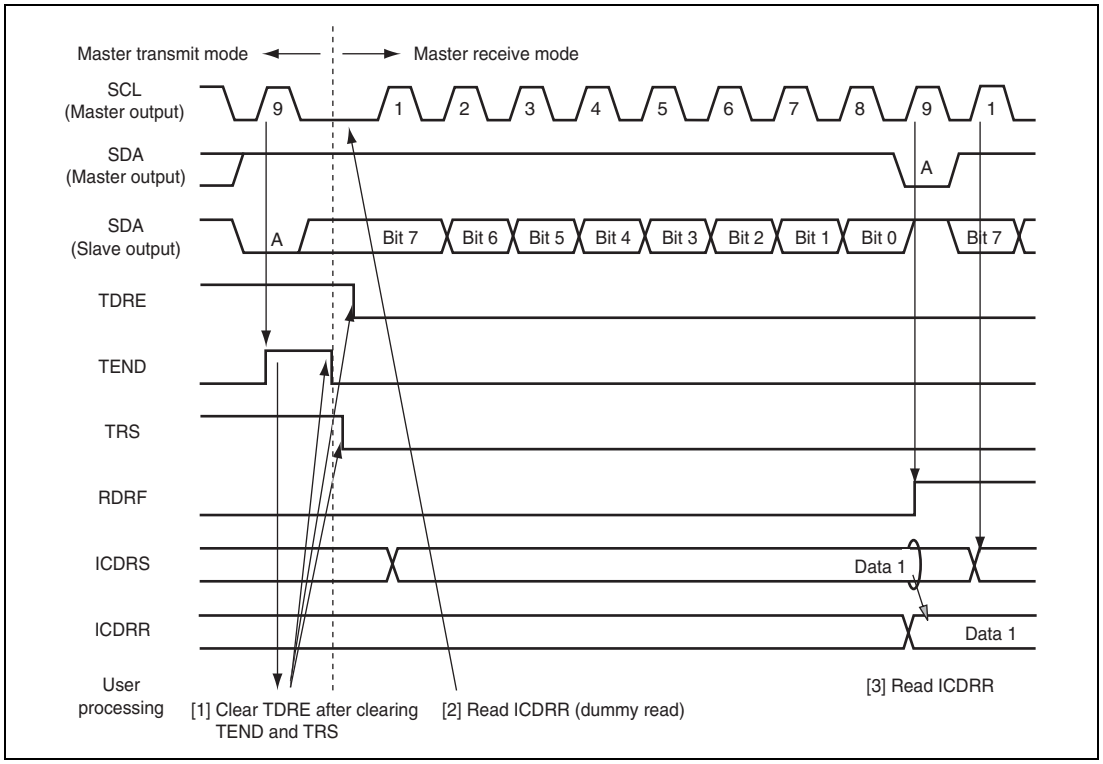


Figure 17.7 Master Receive Mode Operation Timing (1)

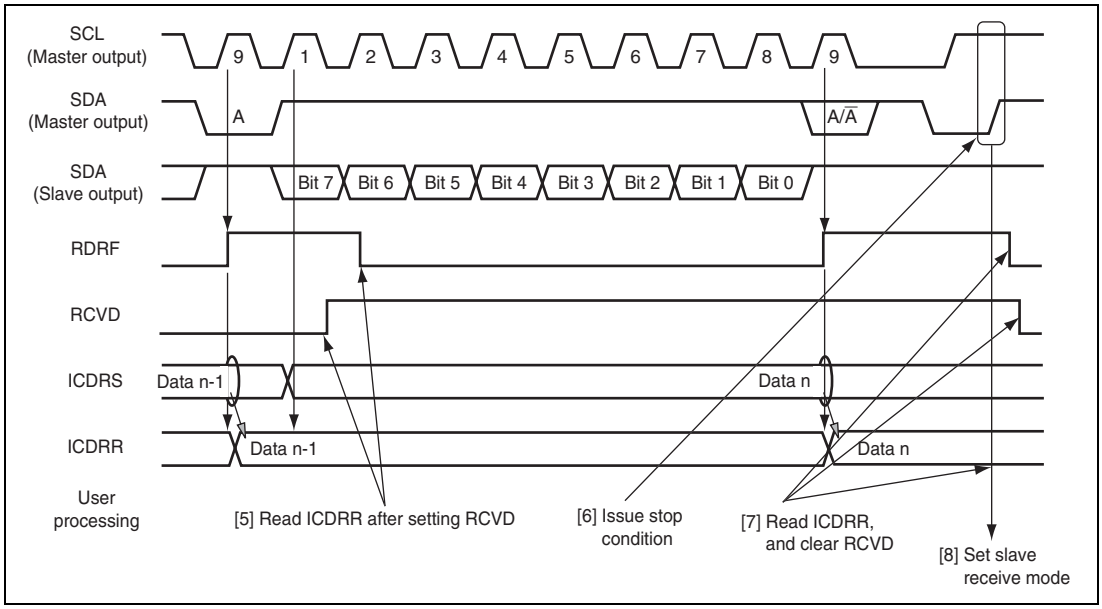


Figure 17.8 Master Receive Mode Operation Timing (2)

17.4.4 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 17.9 and 17.10.

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/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.

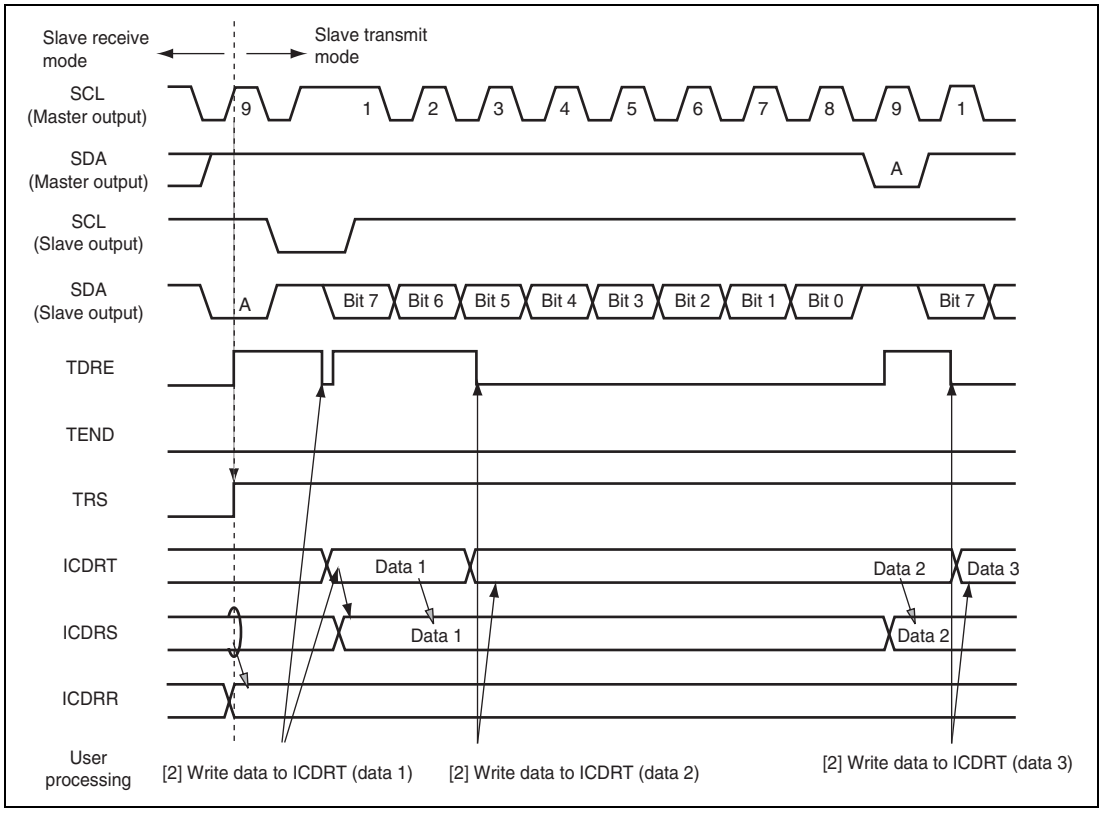


Figure 17.9 Slave Transmit Mode Operation Timing (1)

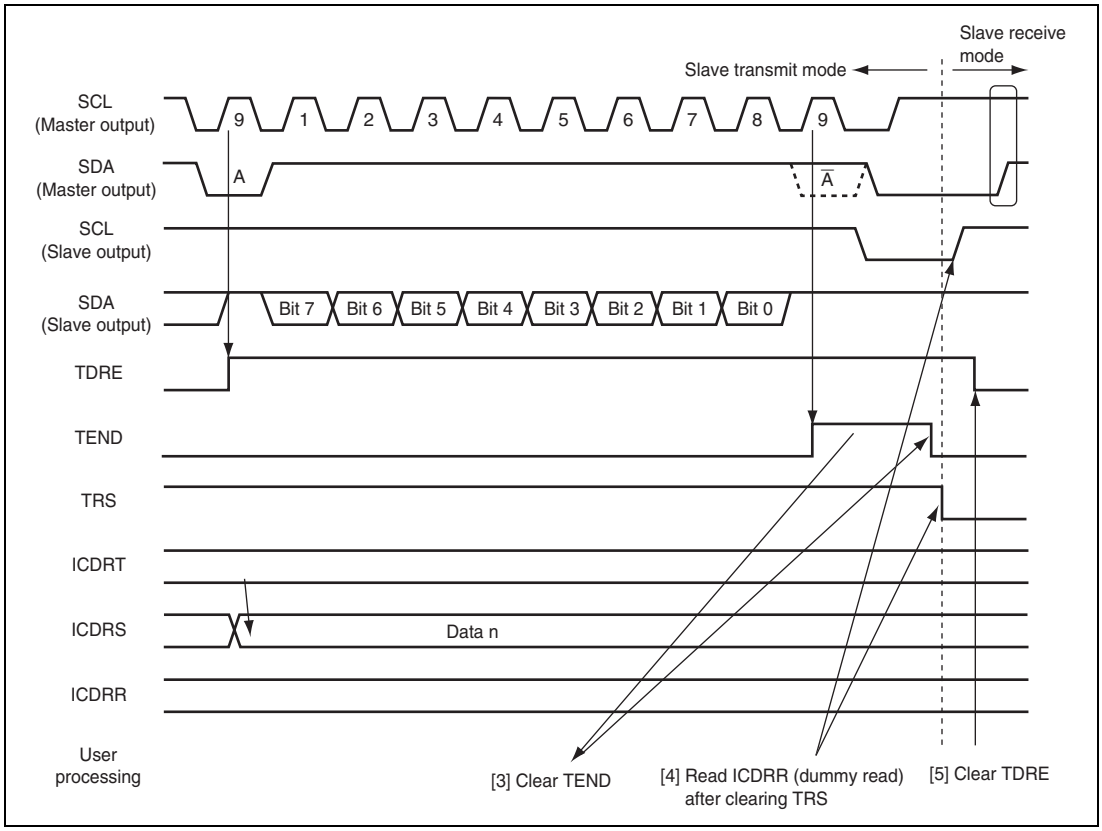


Figure 17.10 Slave Transmit Mode Operation Timing (2)

17.4.5 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 17.11 and 17.12. 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/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.

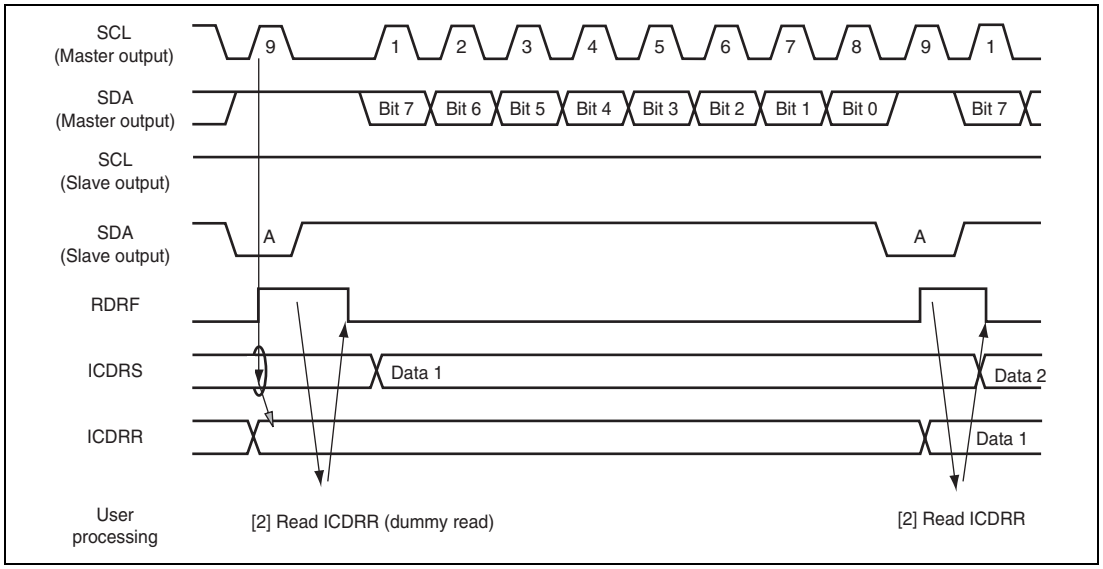


Figure 17.11 Slave Receive Mode Operation Timing (1)

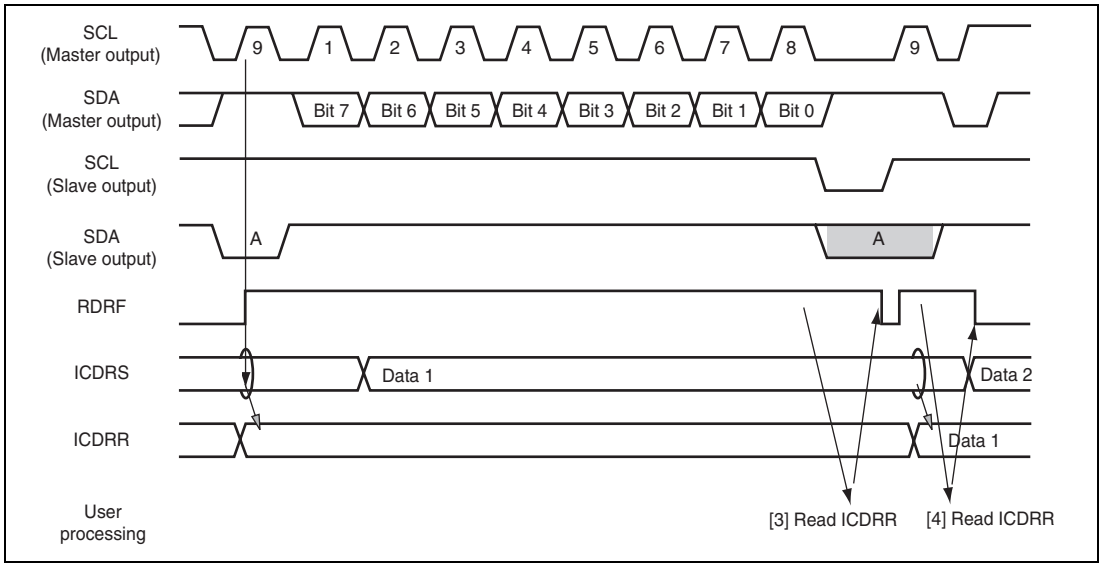


Figure 17.12 Slave Receive Mode Operation Timing (2)

17.4.6 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 17.13 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.

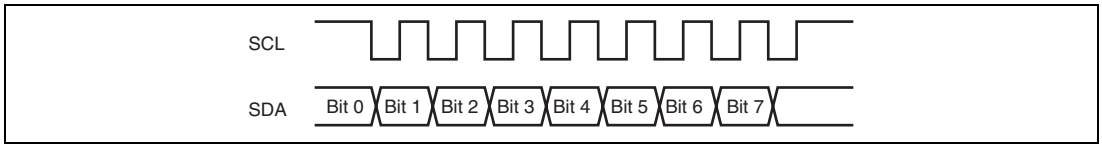


Figure 17.13 Clocked Synchronous Serial Transfer Format

(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 17.14. 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 the 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.

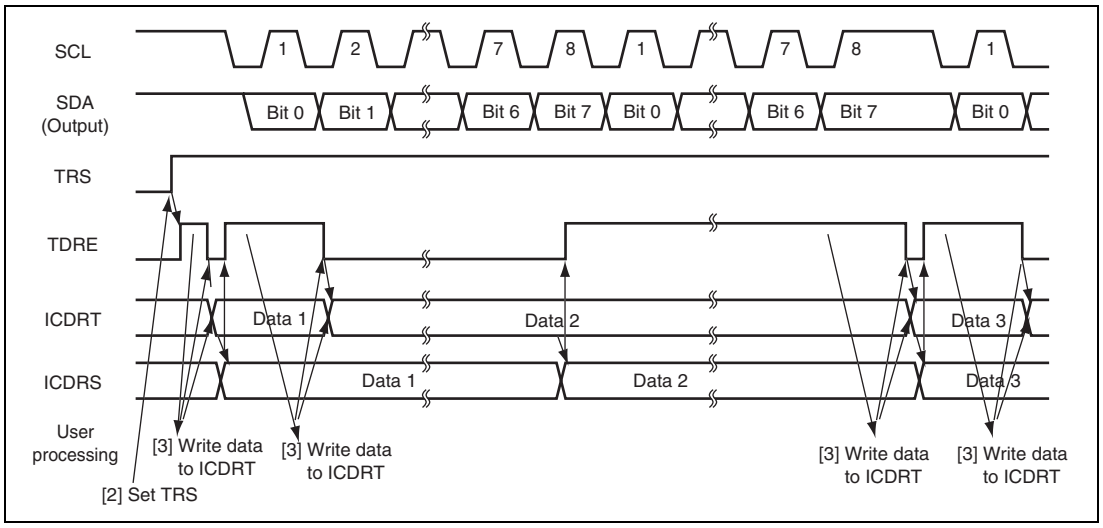


Figure 17.14 Transmit Mode Operation Timing

(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 17.15. 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 17.16 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.

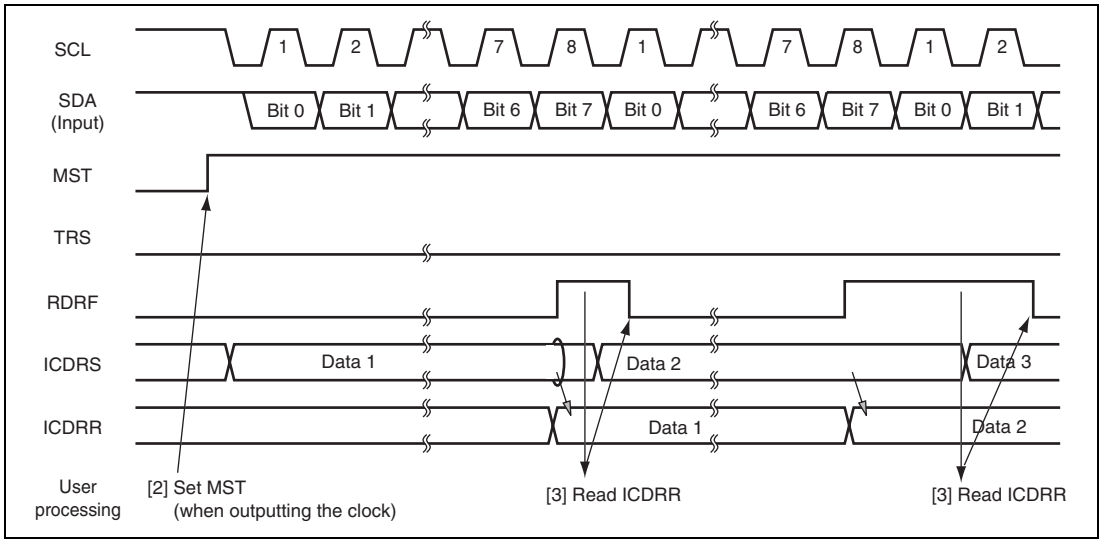


Figure 17.15 Receive Mode Operation Timing

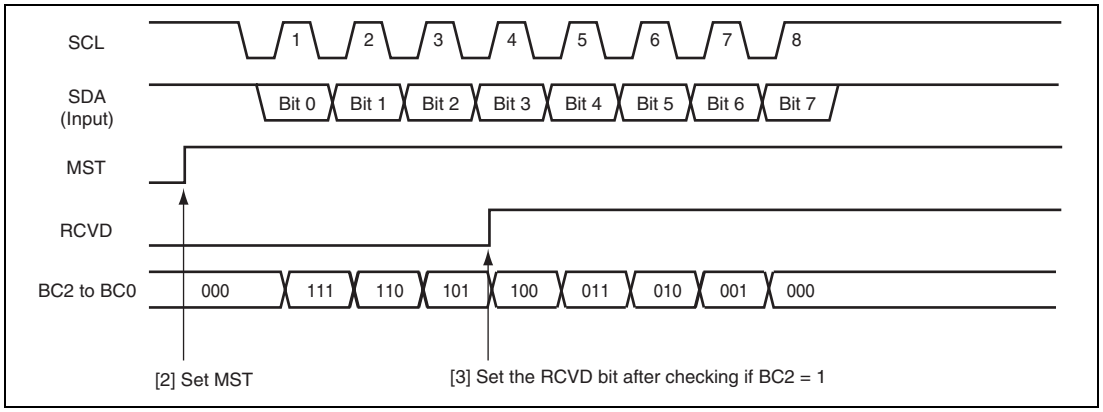


Figure 17.16 Operation Timing For Receiving One Byte (MST = 1)

17.4.7 Noise Filter

The logic levels at the SCL and SDA pins are routed through noise filters before being latched internally. Figure 17.17 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.

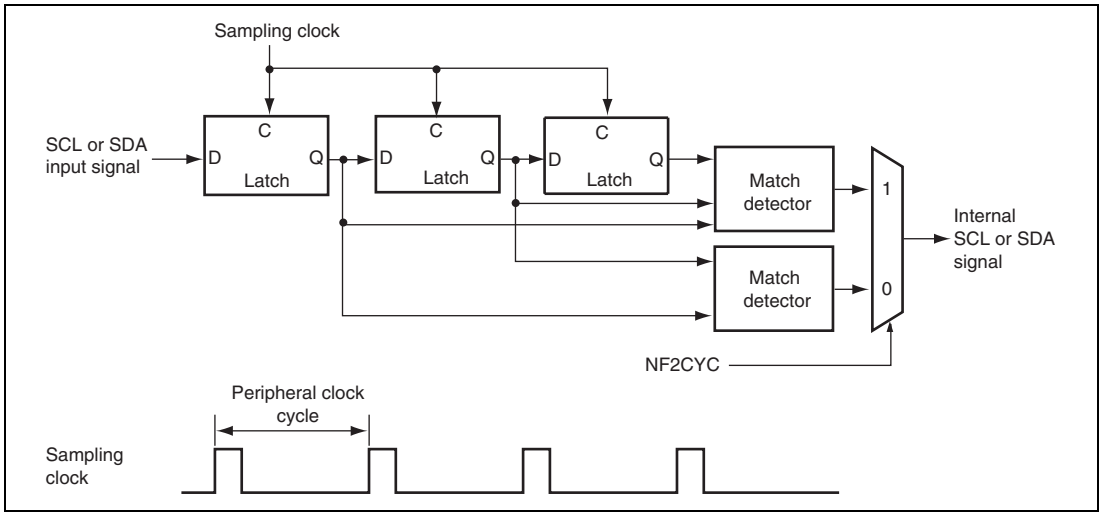


Figure 17.17 Block Diagram of Noise Filter

17.4.8 Example of Use

Flowcharts in respective modes that use the I²C bus interface 3 are shown in figures 17.18 to 17.21.

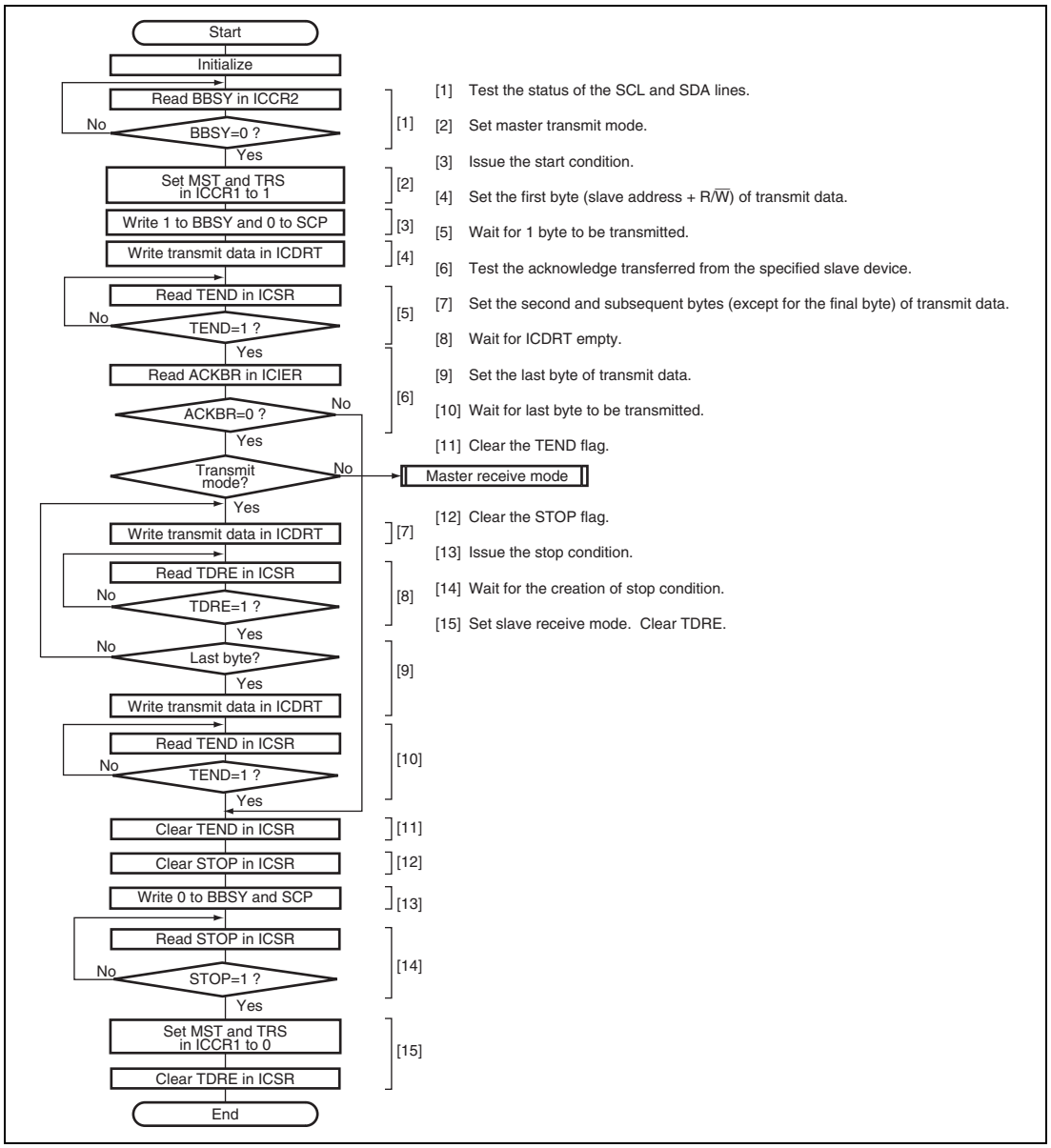


Figure 17.18 Sample Flowchart for Master Transmit Mode

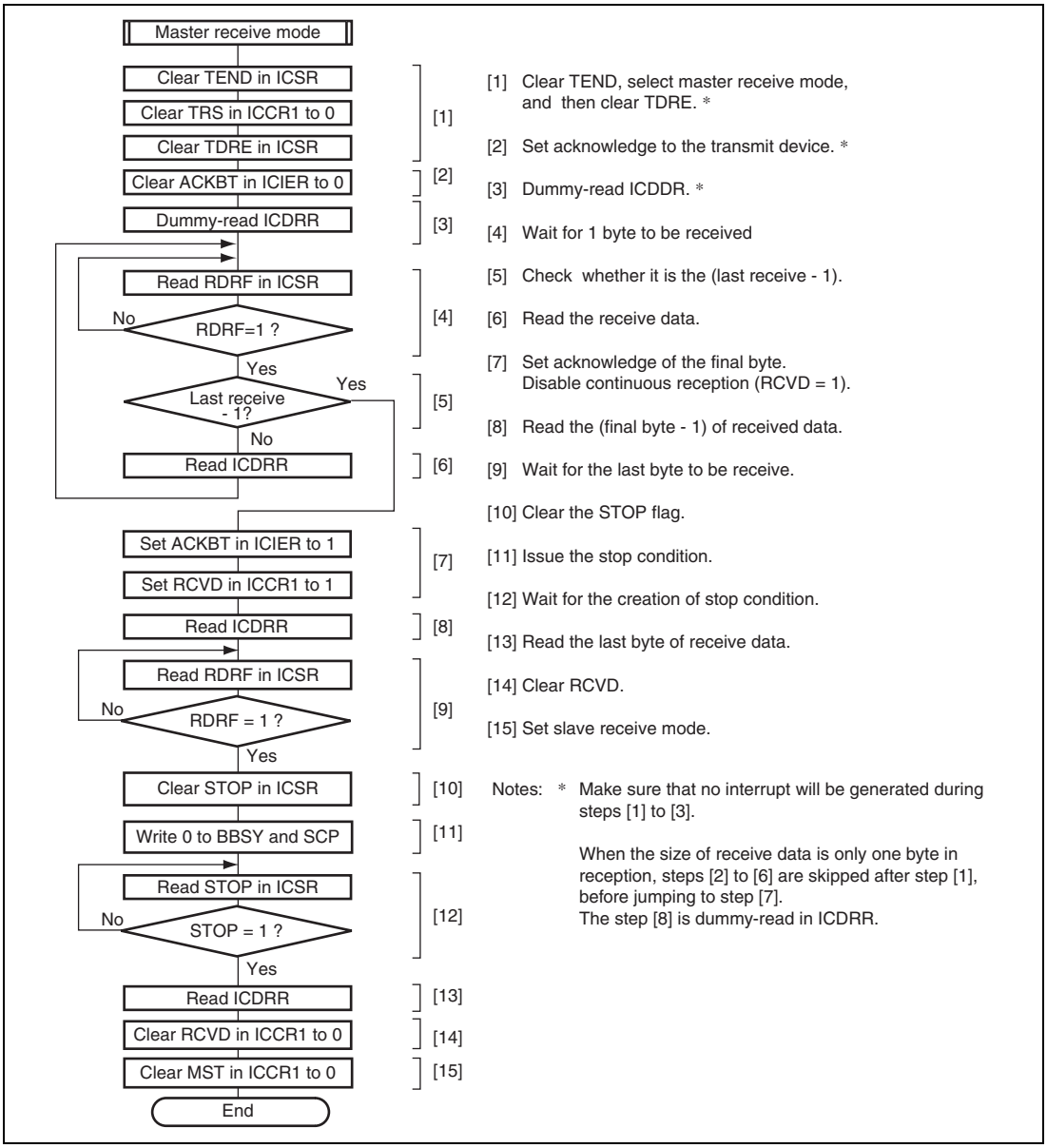


Figure 17.19 Sample Flowchart for Master Receive Mode

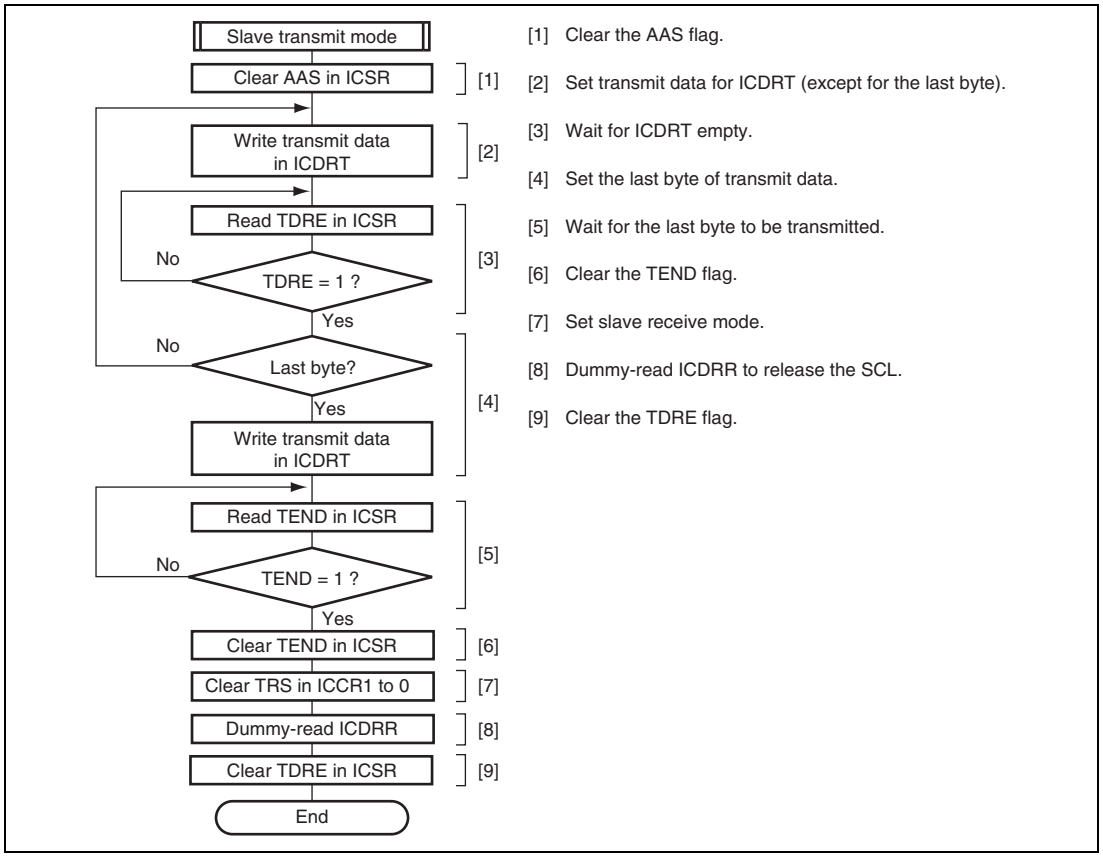


Figure 17.20 Sample Flowchart for Slave Transmit Mode

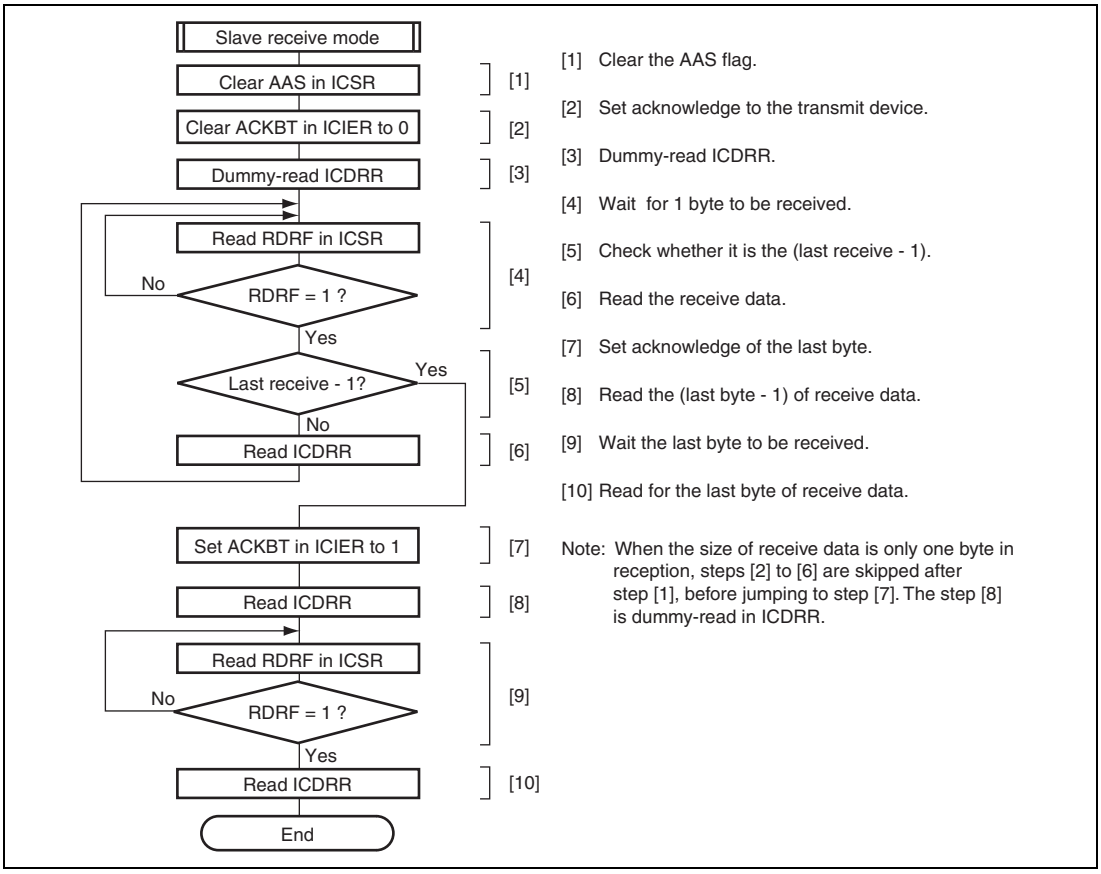


Figure 17.21 Sample Flowchart for Slave Receive Mode

17.5 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 17.4 shows the contents of each interrupt request.

Table 17.4 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 17.4 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.

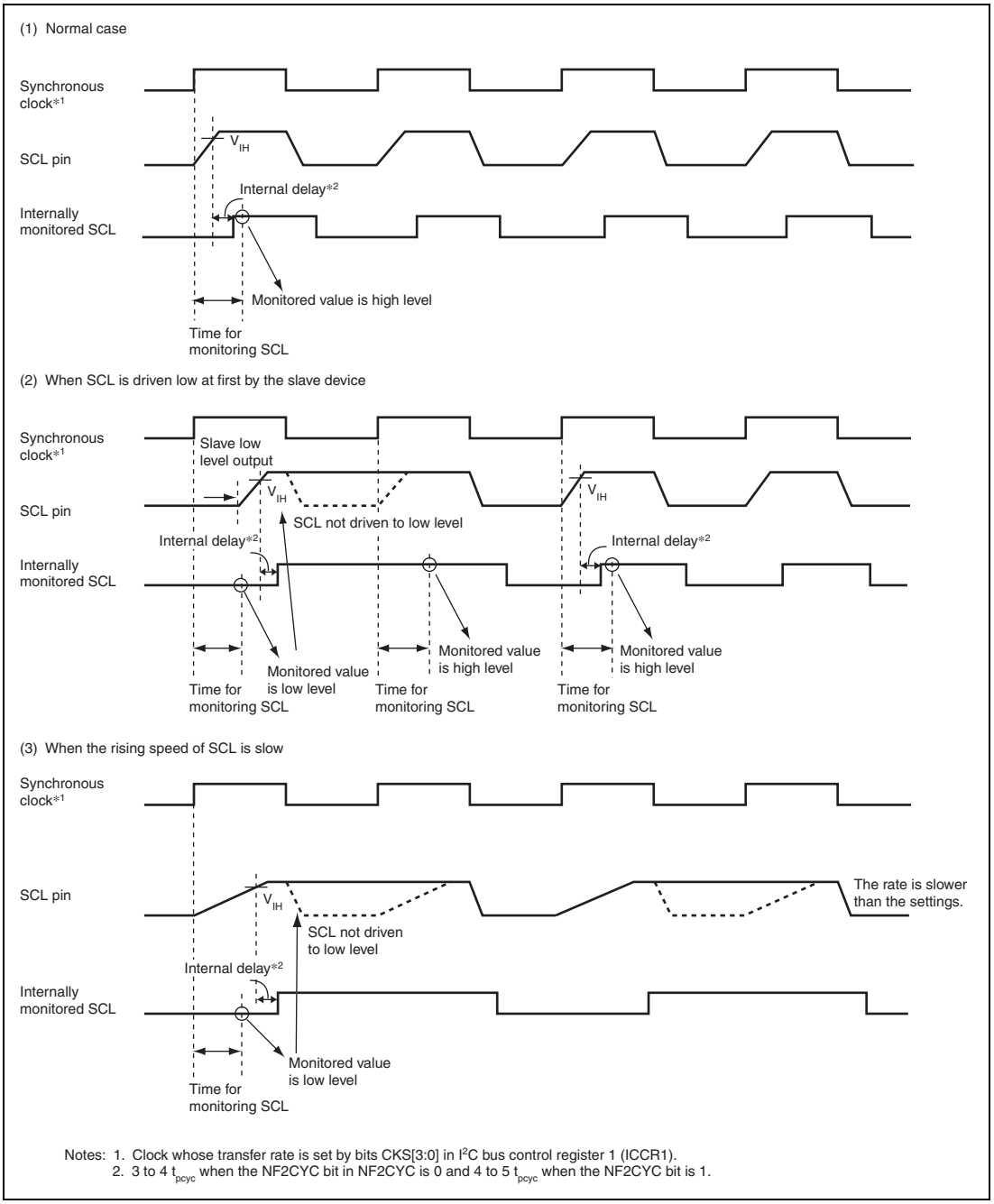
17.6 Bit Synchronous Circuit

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 17.22 shows the timing of the bit synchronous circuit and table 17.5 shows the time when the SCL output changes from low to Hi-Z then SCL is monitored.



Notes: 1. Clock whose transfer rate is set by bits CKS[3:0] in I²C bus control register 1 (ICCR1).
2. 3 to 4 t_{pcyc} when the NF2CYC bit in NF2CYC is 0 and 4 to 5 t_{pcyc} when the NF2CYC bit is 1.

Figure 17.22 Bit Synchronous Circuit Timing

Table 17.5 Time for Monitoring SCL

CKS3	CKS2	Time for Monitoring SCL*[1]
0	0	9 tpcyc*[2]
	1	21 tpcyc*[2]
1	0	39 tpcyc*[2]
	1	87 tpcyc*[2]

Notes: 1. Monitors the (on-board) SCL level after the time (pcyc) for monitoring SCL has passed since the rising edge of the SCL monitor timing reference clock.
2. pcyc = Pϕ × cyc

17.7 Usage Note

17.7.1 Issuance of Stop Condition and Start Condition (Retransmission)

Issue a start (retransmission) or stop condition after the falling edge of the 9th clock has been recognized. The falling edge of the 9th clock can be recognized by checking the SCLO bit in the I²C bus control register 2 (ICCR2). When a start (retransmission) or stop condition is issued with a certain timing under the following conditions (1 or 2), the start (retransmission) or stop condition may not be output correctly.

1. SCL takes longer to rise than the period defined in section 17.6, Bit Synchronous Circuit, due to the load of the SCL bus (load capacitance or pull-up resistance).
2. The low-level period between the 8th and 9th clock is prolonged by the slave device, which activates the bit synchronous circuit.

17.7.2 Note on Setting for Multi-Master Operation

In multi-master operation, when the transfer rate setting for this module (ICCR1.CKS[3:0]) makes this LSI slower than the other masters, pulse cycles with an unexpected length will infrequently be output on SCL.

Be sure to specify a transfer rate that is at least 1/1.8 of the fastest transfer rate among the other masters.

17.7.3 Note on Master Receive Mode

Reading ICDRR around the falling edge of the 8th clock might fail to fetch the receive data.

In addition, when RCVD is set to 1 around the falling edge of the 8th clock and the receive buffer full, a stop condition may not be issued.

Use either 1 or 2 below as a measure against the situations above.

1. In master receive mode, read ICDRR before the rising edge of the 8th clock.
2. In master receive mode, set the RCVD bit to 1 so that transfer proceeds in byte units.

17.7.4 Note on Setting ACKBT in Master Receive Mode

In master receive mode operation, set ACKBT before the falling edge of the 8th SCL cycle of the last data being continuously transferred. Not doing so can lead to an overrun for the slave transmission device.

17.7.5 Note on the States of Bits MST and TRN when Arbitration is Lost

When sequential bit-manipulation instructions are used to set the MST and TRS bits to select master transmission in multi-master operation, a conflicting situation where AL in ICSR = 1 but the mode is master transmit mode (MST = 1 and TRS = 1) may arise; this depends on the timing of the loss of arbitration when the bit manipulation instruction for TRS is executed.

This can be avoided in either of the following ways.

- In multi-master operation, use the MOV instruction to set the MST and TRS bits.
- When arbitration is lost, check whether the MST and TRS bits are 0. If the MST and TRS bits have been set to a value other than 0, clear the bits to 0.

17.7.6 Note on IICRST and BBSY bits

When 1 is written to IICRST in ICCR2, this LSI release SCL and SDA pins. Then, if the SDA level changes from low to high under the condition of SCL = high, BBSY in ICCR2 is cleared to 0 assuming that the stop condition has been issued.

Section 18 Serial Sound Interface (SSI)

The serial sound interface (hereinafter referred to as the "SSI") is a transceiver module designed to send or receive audio data interface with a variety of devices compatible I^2S bus. It also provides additional modes for other common formats as well as multi-channel mode.

18.1 Features

- Number of channels: Two channels
- Operating mode: Non-compressed mode
  - The non-compressed mode supports serial audio streams divided by channels.
- Serves as both a transmitter and a receiver
- Capable of using serial bus format
- Asynchronous transfer takes place between the data buffer and the shift register.
- It is possible to select a value as the dividing ratio for the clock used by the serial but interface.
- It is possible to control data transmission or reception with DMAC and interrupt requests.
- Selects the oversample clock from among the pins AUDIO_CLK, or AUDIO_X1 and AUDIO_X2.
  - External clock frequency input through the pins AUDIO_CLK, or AUDIO_X1 and AUDIO_X2: 1 to 40 MHz
  - Crystal oscillator frequency for the pins AUDIO_X1 and AUDIO_X2: 10 to 25 MHz

Figure 18.1 shows a schematic diagram of the four channels in the SSI module.

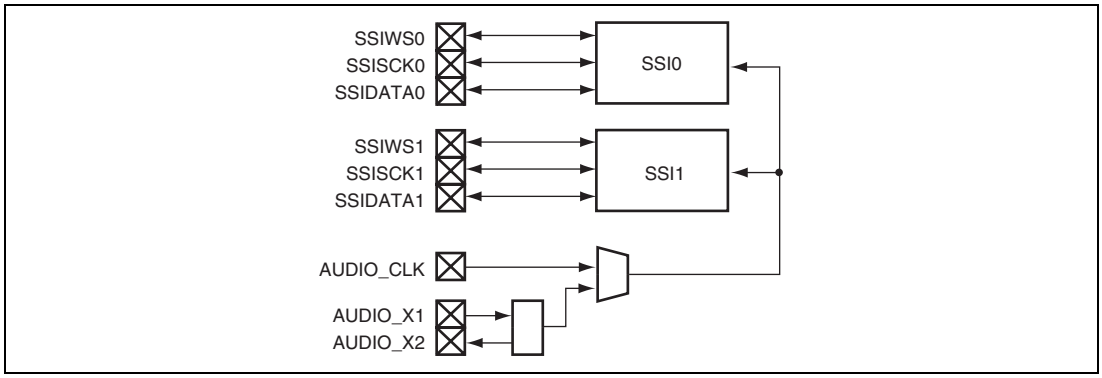


Figure 18.1 Schematic Diagram of SSI Module

Figure 18.2 shows a block diagram of the SSI module when it is used alone.

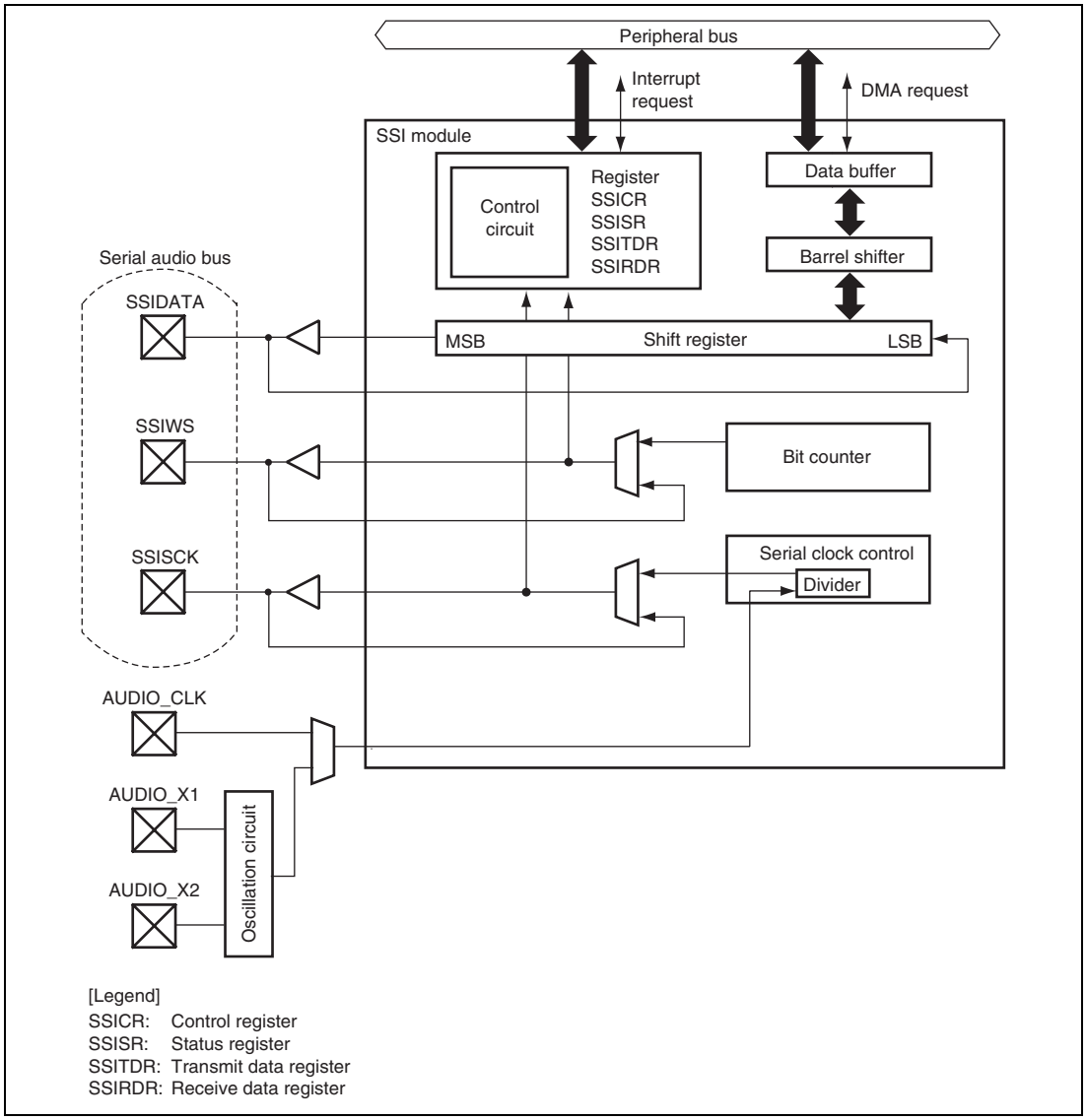


Figure 18.2 Block Diagram of SSI

18.2 Input/Output Pins

Table 18.1 shows the pin assignments relating to the SSI module.

Table 18.1 Pin Assignments

Pin Name	Number of Pins	I/O	Description
SSISCK0	1	I/O	Serial bit clock
SSIWS0	1	I/O	Word selection
SSIDATA0	1	I/O	Serial data input/output
SSISCK1	1	I/O	Serial bit clock
SSIWS1	1	I/O	Word selection
SSIDATA1	1	I/O	Serial data input/output
AUDIO_CLK	1	Input	External clock for audio (Oversample clock)
AUDIO_X1	1	Input	Crystal oscillator for audio (Oversample clock)
AUDIO_X1	1	Output	

18.3 Register Description

The SSI has the following registers. Note that explanation in the text does not refer to the channels.

Table 18.2 Register Description

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	Control register 0	SSICR0	R/W	H'00000000	H'FFFED000	32
	Status register 0	SSISR0	R/W*	H'02000003	H'FFFED004	32
	Transmit data register 0	SSITDR0	R/W	H'00000000	H'FFFED008	32
	Receive data register 0	SSIRDR0	R	H'00000000	H'FFFED00C	32
1	Control register 1	SSICR1	R/W	H'00000000	H'FFFED080	32
	Status register 1	SSISR1	R/W*	H'02000003	H'FFFED084	32
	Transmit data register 1	SSITDR1	R/W	H'00000000	H'FFFED088	32
	Receive data register 1	SSIRDR1	R	H'00000000	H'FFFED08C	32

Note: * For this register, bits 26 and 27 are capable of reading and writing, although the others are read-only bits. For details, refer to section 18.3.2, Status Register (SSISR).

18.3.1 Control Register (SSICR)

SSICR is a readable/writable 32-bit register that controls the IRQ, selects the polarity status, and sets operating mode.

SSICR is initialized to H'00000000 by a power-on reset or in deep standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL[1:0]		DWL[2:0]			SWL[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/W	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
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	—	CKDV[2:0]			MUEN	—	TRMD	EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved The read value is not guaranteed. The write value should always be 0.
28	DMEN	0	R/W	DMA Enable Enables/disables the DMA request. 0: DMA request is disabled. 1: DMA request is enabled.
27	UIEN	0	R/W	Underflow Interrupt Enable 0: Underflow interrupt is disabled. 1: Underflow Interrupt is enabled.
26	OIEN	0	R/W	Overflow Interrupt Enable 0: Overflow interrupt is disabled. 1: Overflow interrupt is enabled.
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Idle mode interrupt is disabled. 1: Idle mode interrupt is enabled.
24	DIEN	0	R/W	Data Interrupt Enable 0: Data interrupt is disabled. 1: Data interrupt is enabled.

Bit	Bit Name	Initial Value	R/W	Description
23, 22	CHNL[1:0]	00	R/W	Channels These bits show the number of channels in each System Word. 00: Having one channel per System Word 01: Having two channels per System Word 10 Having three channels per System Word 11: Having four channels per System Word
21 to 19	DWL[2:0]	000	R/W	Data Word Length Indicates the number of bits in a data word. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 22 bits 101: 24 bits 110: 32 bits 111: Reserved
18 to 16	SWL[2:0]	000	R/W	System Word Length Indicates the number of bits in a system word. 000: 8 bits 001: 16 bits 010: 24 bits 011: 32 bits 100: 48 bits 101: 64 bits 110: 128 bits 111: 256 bits
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial bit clock is input, slave mode. 1: Serial bit clock is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.

Bit	Bit Name	Initial Value	R/W	Description
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select is input, slave mode. 1: Serial word select is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SCK falling edge).
12	SWSP	0	R/W	Serial WS Polarity 0: SSIWS is low for 1st channel, high for 2nd channel. 1: SSIWS is high for 1st channel, low for 2nd channel.
11	SPDP	0	R/W	Serial Padding Polarity 0: Padding bits are low. 1: Padding bits are high. Note: When MUEN = 1, padding bits are low. (The MUTE function is given priority.)
10	SDTA	0	R/W	Serial Data Alignment 0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data

	SCKP = 0	SCKP = 1
SSIDATA input sampling timing at the time of reception (TRMD = 0)	SSISCK rising edge	SSISCK falling edge
SSIDATA output change timing at the time of transmission (TRMD = 1)	SSISCK falling edge	SSISCK rising edge
SSIWS input sampling timing at the time of slave mode (SWSD = 0)	SSISCK rising edge	SSISCK falling edge
SSIWS output change timing at the time of master mode (SWSD = 1)	SSISCK falling edge	SSISCK rising edge

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	Parallel Data Alignment This bit is ignored if CPEN = 1. When the data word length is 32, 16 or 8 bit, this configuration field has no meaning. This bit applies to SSIRDR in receive mode and SSITDR in transmit mode. 0: Parallel data (SSITDR, SSIRDR) is left-aligned 1: Parallel data (SSITDR, SSIRDR) is right-aligned. • DWL = 000 (with a data word length of 8 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Four data words are transmitted or received at each 32-bit access. The first data word is derived from bits 7 to 0, the second from bits 15 to 8, the third from bits 23 to 16 and the last data word is derived from bits 31 to 24. • DWL = 001 (with a data word length of 16 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus. Two data words are transmitted or received at each 32-bit access. The first data word is derived from bits 15 to 0 and the second data word is derived from bits 31 to 16. • DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 0 (left-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits 31 down to (32 minus the number of bits in the data word length specified by DWL). That is, If DWL = 011, the data word length is 20 bits; therefore, bits 31 to 12 in either SSIRDR or SSITDR are used. All other bits are ignored or reserved.

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• DWL = 010, 011, 100, 101 (with a data word length of 18, 20, 22 or 24 bits), PDTA = 1 (right-aligned) The data bits used in SSIRDR or SSITDR are the following: Bits (the number of bits in the data word length specified by DWL minus 1) to 0 i.e. if DWL = 011, then DWL = 20 and bits 19 to 0 are used in either SSIRDR or SSITDR. All other bits are ignored or reserved. • DWL = 110 (with a data word length of 32 bits), the PDTA setting is ignored. All data bits in SSIRDR or SSITDR are used on the audio serial bus.
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSIWS and SSIDATA 1: No delay between SSIWS and SSIDATA
7	—	0	R	Reserved The read value is undefined. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
6 to 4	CKDV[2:0]	000	R/W	Serial Oversample Clock Divide Ratio Sets the ratio between oversample clock* (AUDIO_CLK, or AUDIO_X1 and AUDIO_X2) and the serial bit clock. In addition, combining these bits and the CKDV3 bit in the standby control register enables to divide the clock further by 1/4. This bit is ignored if SCKD = 0. The serial bit clock is used in the shift register and is provided on the SSISCK module pin. • When CKDV3 = 1 000: Serial bit clock frequency = Oversample clock Frequency/1 001: Serial bit clock frequency = Oversample clock frequency/2 010: Serial bit clock frequency = Oversample clock frequency/4 011: Serial bit clock frequency = Oversample clock frequency/8 100: Serial bit clock frequency = Oversample clock frequency/16 101: Serial bit clock frequency = Oversample clock frequency/6 110: Serial bit clock frequency = Oversample clock frequency/12 111: Setting prohibited • When CKDV3 = 0 000: Serial bit clock frequency = Oversample clock Frequency/4 001: Serial bit clock frequency = Oversample clock frequency/8 010: Serial bit clock frequency = Oversample clock frequency/16 011: Serial bit clock frequency = Oversample clock frequency/32 100: Serial bit clock frequency = Oversample clock frequency/64 101: Serial bit clock frequency = Oversample clock frequency/24 110: Serial bit clock frequency = Oversample clock frequency/48 111: Setting prohibited Note: * AUDIO_X1 and AUDIO_X2 is selected as oversample clock when the PD0MD0 bit in the port D control register (PDCR1) of PFC is set to 0, and AUDIO_CLK is selected when the bit is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	MUEN	0	R/W	Mute Enable 0: Module is not muted. 1: Module is muted.
2	—	0	R	Reserved The read value is undefined. The write value should always be 0.
1	TRMD	0	R/W	Transmit/Receive Mode Select 0: Module is in receive mode. 1: Module is in transmit mode.
0	EN	0	R/W	SSI Module Enable 0: Module is disabled. 1: Module is enabled.

18.3.2 Status Register (SSISR)

SSISR consists of status flags indicating the operational status of the SSI module and bits indicating the current channel numbers and word numbers.

SSISR is initialized to H'02000003 by a power-on reset or in deep standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	1*2	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W*1	R/W*1	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
	—	—	—	—	—	—	—	—	—	—	—	—	CHNO[1:0]		SWNO	IDST
Initial value:	—	—	—	—	—	—	—	—	—	—	—	—	0	0	1	1*2
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Notes: 1. This bit can be read from or written to. Writing 0 initializes the bit, but writing 1 is ignored.
2. The SSI clock must be kept supplied until the SSI is in the idle state.

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved The read value is not guaranteed. The write value should always be 0.
28	DMRQ	0	R	DMA Request Status Flag This status flag allows the CPU to recognize the value of the DMA request pin on the SSI module. • TRMD = 0 (Receive mode) If DMRQ = 1, the SSIRDR has unread data. If SSIRDR is read, DMRQ = 0 until there is new unread data. • TRMD = 1 (Transmit mode) If DMRQ = 1, SSITDR requires data to be written to continue the transmission to the audio serial bus. Once data is written to SSITDR, DMRQ = 0 until it requires further transmit data.
27	UIRQ	0	R/W*[1]	Underflow Error Interrupt Status Flag This status flag indicates that data was supplied at a lower rate than was required. In either case, this bit is set to 1 regardless of the value of the UIEN bit and can be cleared by writing 0 to this bit. If UIRQ = 1 and UIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) If UIRQ = 1, SSIRDR was read before there was new unread data indicated by the DMRQ or DIRQ bit. This can lead to the same received sample being stored twice by the host leading to potential corruption of multi-channel data. • TRMD = 1 (Transmit mode) If UIRQ = 1, SSITDR did not have data written to it before it was required for transmission. This will lead to the same sample being transmitted once more and a potential corruption of multi-channel data. This is more serious error than a receive mode underflow as the output SSI data results in error. Note: When underflow error occurs, the current data in the data buffer of this module is transmitted until the next data is filled.

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/W*[1]	Overflow Error Interrupt Status Flag This status flag indicates that data was supplied at a higher rate than was required. In either case this bit is set to 1 regardless of the value of the OIEN bit and can be cleared by writing 0 to this bit. If OIRQ = 1 and OIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) If OIRQ = 1, SSIRDR was not read before there was new unread data written to it. This will lead to the loss of a sample and a potential corruption of multi-channel data. Note: When overflow error occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSI interface. • TRMD = 1 (Transmit mode) If OIRQ = 1, SSITDR had data written to it before it was transferred to the shift register. This will lead to the loss of a sample and a potential corruption of multi-channel data.
25	IIRQ	1*[2]	R	Idle Mode Interrupt Status Flag This interrupt status flag indicates whether the SSI module is in idle state. This bit is set regardless of the value of the IIEN bit to allow polling. The interrupt can be masked by clearing IIEN, but cannot be cleared by writing to this bit. If IIRQ = 1 and IIEN = 1, an interrupt occurs. 0: The SSI module is not in idle state. 1: The SSI module is in idle state.

Bit	Bit Name	Initial Value	R/W	Description
24	DIRQ	0	R	Data Interrupt Status Flag This status flag indicates that the module has data to be read or requires data to be written. In either case this bit is set to 1 regardless of the value of the DIEN bit to allow polling. The interrupt can be masked by clearing DIEN, but cannot be cleared by writing to this bit. If DIRQ= 1 and DIEN = 1, an interrupt occurs. • TRMD = 0 (Receive mode) 0: No unread data in SSIRDR 1: Unread data in SSIRDR • TRMD = 1 (Transmit mode) 0: Transmit buffer is full. 1: Transmit buffer is empty and requires data to be written to SSITDR.
23 to 4	—	Undefined	R	Reserved The read value is not guaranteed. The write value should always be 0.
3, 2	CHNO[1:0]	00	R	Channel Number This value indicates the current channel number. • TRMD = 0 (Receive mode) CHNO indicates which channel the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register. • TRMD = 1 (Transmit mode) CHNO indicates which channel is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR.

Bit	Bit Name	Initial Value	R/W	Description
1	SWNO	1	R	System Word Number This status bit indicates the current word number. • TRMD = 0 (Receive mode) SWNO indicates which system word the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register, regardless of whether SSIRDR has been read. • TRMD = 1 (Transmit mode) SWNO indicates which system word is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR.
0	IDST	1*[2]	R	Idle Mode Status Flag This status flag indicates that the serial bus activity has stopped. This bit is cleared if EN = 1 and the serial bus are currently active. This bit is automatically set to 1 under the following conditions. • SSI = Master transmitter (SWSD = 1 and TRMD = 1) This bit is set to 1 if the EN bit is cleared and the data written to SSITDR is completely output from the serial data input/output pin (SSIDATA), that is, the output of the system word length is completed. • SSI = Master receiver (SWSD = 1 and TRMD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed. • SSI = Slave transmitter/receiver (SWSD = 0) This bit is set to 1 if the EN bit is cleared and the current system word is completed. Note: If the external master stops the serial bus clock before the current system word is completed, this bit is not set.

Notes: 1. This bit can be read from or written to. Writing 0 initializes the bit, but writing 1 is ignored.
2. The SSI clock must be kept supplied until the SSI is in the idle state.

18.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted.

Data written to this register is transferred to the shift register upon transmission request. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR. The data in the buffer can be accessed by reading this register.

SSITDR is initialized to H'00000000 by a power-on reset or in deep 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

18.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores receive messages.

Data in this register is transferred from the shift register each time data word is received. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

SSIRDR is initialized to H'00000000 by a power-on reset or in deep 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	R	R	R	R	R	R	R

18.4 Operation Description

18.4.1 Bus Format

The SSI module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus format can be selected from one of the four major modes shown in table 18.3.

Table 18.3 Bus Format for SSI Module

	Non-Compressed Slave Receiver	Non-Compressed Slave Transmitter	Non-Compressed Master Receiver	Non-Compressed Master Transmitter
TRMD	0	1	0	1
SCKD	0	0	1	1
SWSD	0	0	1	1
EN	Control Bits			
MUEN				
DIEN				
IIEN				
OIEN				
UIEN				
DEL	Configuration Bits			
PDTA				
SDTA				
SPDP				
SWSP				
SCKP				
SWL [2:0]				
DWL [2:0]				
CHNL [1:0]				

18.4.2 Non-Compressed Modes

The non-compressed modes support all serial audio streams split into channels. It supports I^2S compatible format as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of the SSI module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of the SSI module, operation is not guaranteed.

(3) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of the SSI module. If the incoming data does not follow the configured format, operation is not guaranteed.

(4) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of the SSI module.

(5) Operating Setting Related to Word Length

All bits related to the SSICR's word length are valid in non-compressed modes. The SSI module supports many configurations, but the formats described below are I^2S compatible, MSB-first left-aligned, and MSB-first right-aligned.

1. I^2S Compatible Format

Figures 18.3 and 18.4 demonstrate the supported I^2S compatible format both with and without padding. Padding occurs when the data word length is smaller than the system word length.

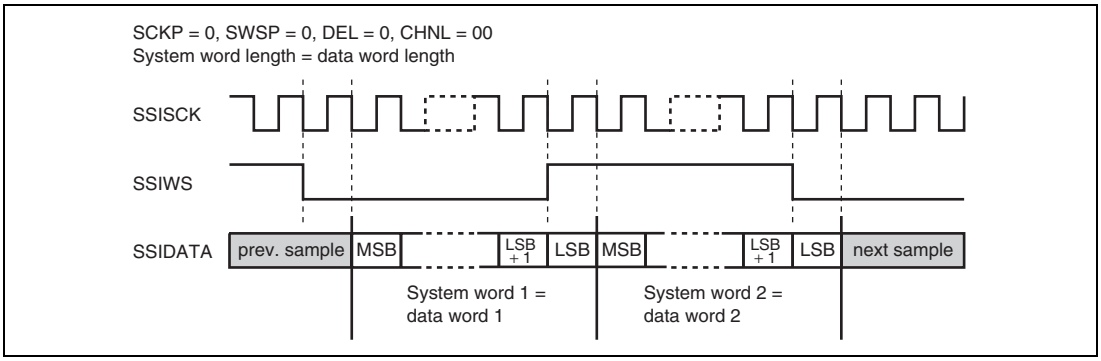


Figure 18.3 I^2S Compatible Format (without Padding)

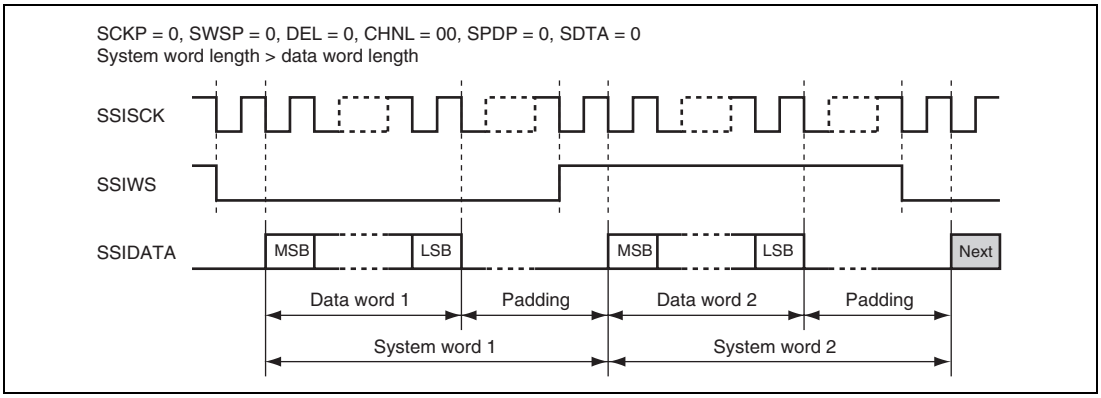


Figure 18.4 I^2S Compatible Format (with Padding)

Figure 18.5 shows MSB-first left-aligned format, and figure 18.6 shows MSB-first right-aligned format.

2. MSB-First Left-Aligned Format

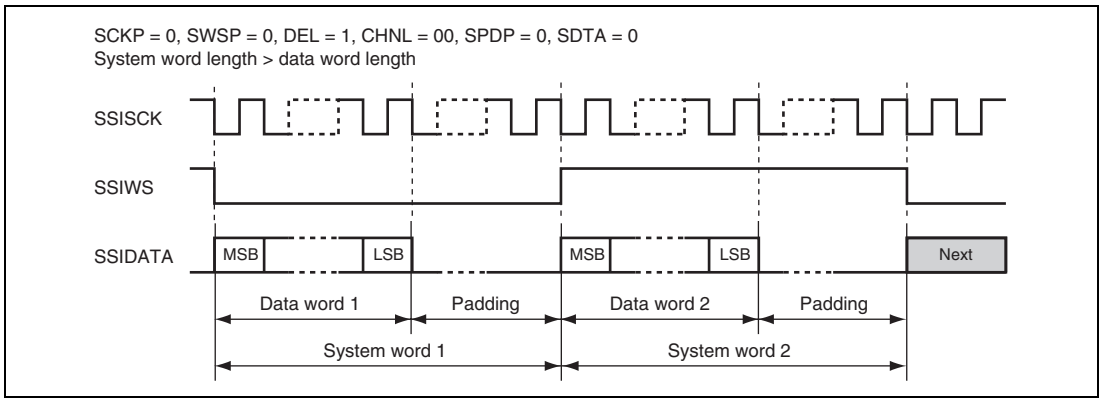


Figure 18.5 MSB-First Left-Aligned Format
(Transmitted and Received in the order of Serial Data and Padding Bits)

3. MSB-First Right-Aligned Format

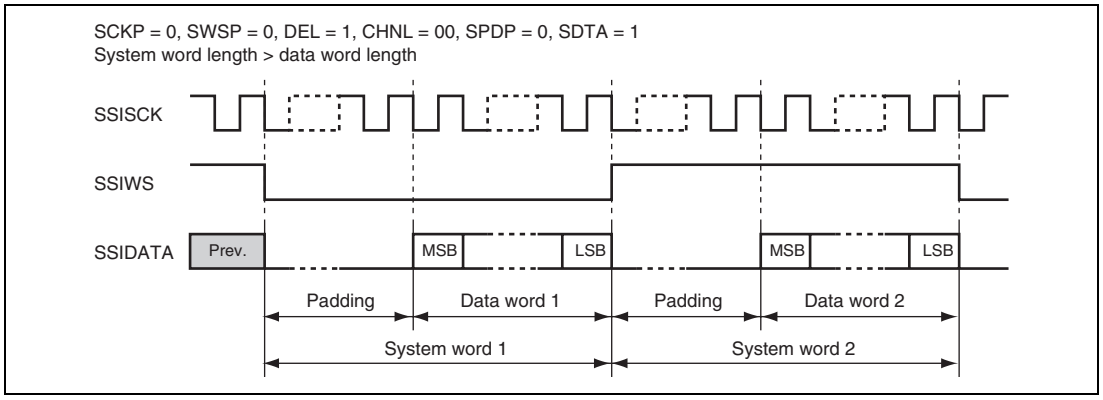


Figure 18.6 MSB-First Right-Aligned Format
(Transmitted and Received in the order of Padding Bits and Serial Data)

(6) Multi-channel Formats

Some devices extend the definition of the specification by I^2S bus and allow more than 2 channels to be transferred within two system words.

The SSI module supports the transfer of 4, 6 and 8 channels by using the CHNL, SWL and DWL bits only when the system word length (SWL) is greater than or equal to the data word length (DWL) multiplied by channels (CHNL).

Table 18.4 shows the number of padding bits for each of the valid setting. If setting is not valid, "—" is indicated instead of a number.

Table 18.4 The Number of Padding Bits for Each Valid Setting

Padding Bits Per System Word			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192

Padding Bits Per System Word			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

When the SSI module acts as a transmitter, each word written to SSITDR is transmitted to the serial audio bus in the order they are written. When the SSI module acts as a receiver, each word received by the serial audio bus is read in the order received from the SSIRDR register.

Figures 18.7 to 18.9 show how 4, 6 and 8 channels are transferred to the serial audio bus. Note that there are no padding bits in the first example, the second example is left-aligned and the third is right-aligned. This selection is arbitrary and is just for demonstration purposes only.

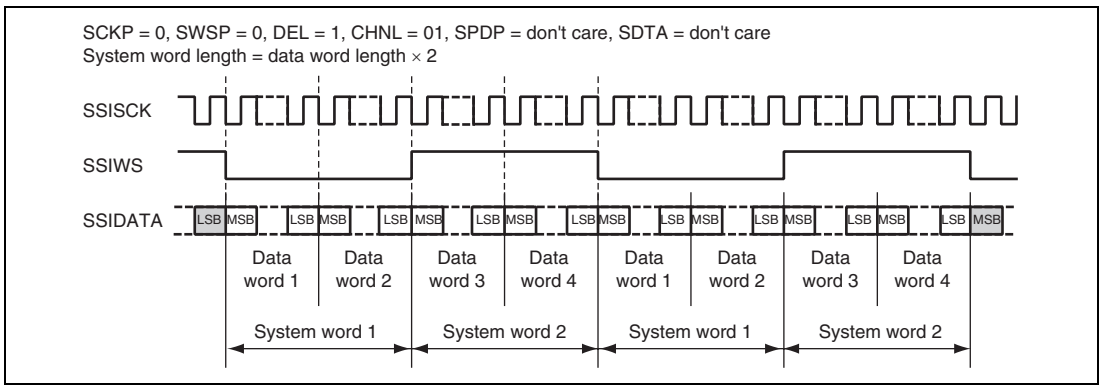


Figure 18.7 Multichannel Format (4 Channels Without Padding)

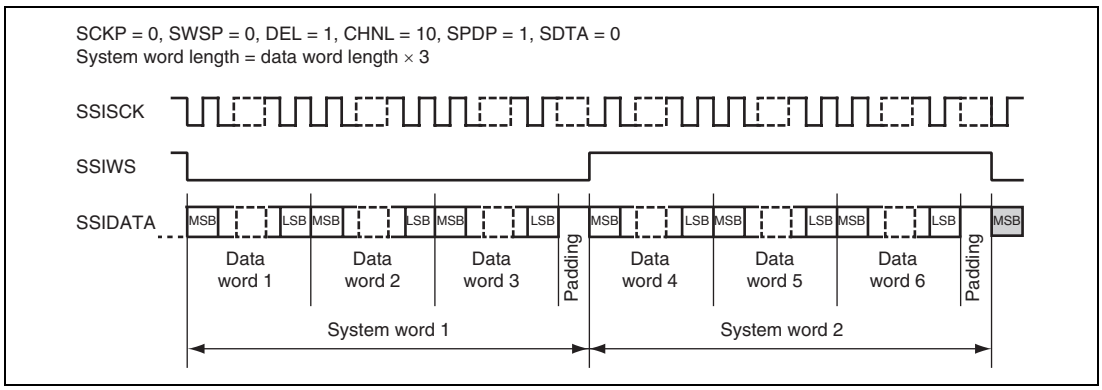


Figure 18.8 Multichannel Format (6 Channels with High Padding)

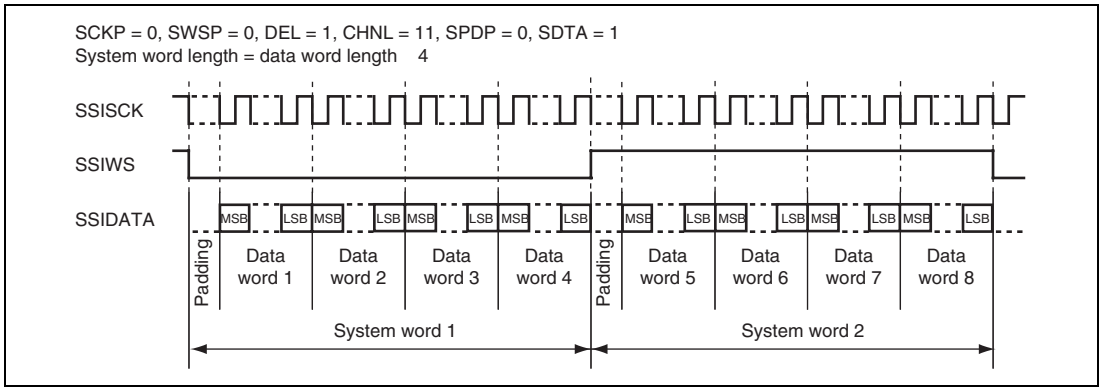


Figure 18.9 Multichannel Format (8 Channels; Transmitting and Receiving in the order of Padding Bits and Serial Data; with Padding)

(7) Bit Setting Configuration Format

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful for any other device.

These configuration bits are described below with reference to figure 18.10, Basic Sample Format.

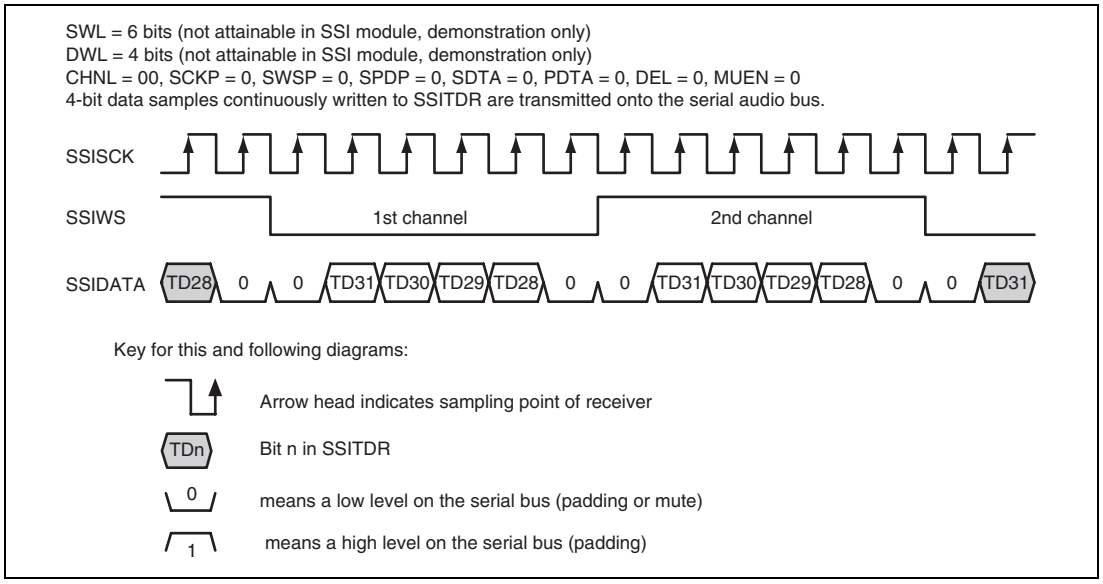


Figure 18.10 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

Figure 18.10 uses a system word length of 6 bits and a data word length of 4 bits. These settings are not possible with the SSI module but are used only for clarification of the other configuration bits.

1. Inverted Clock

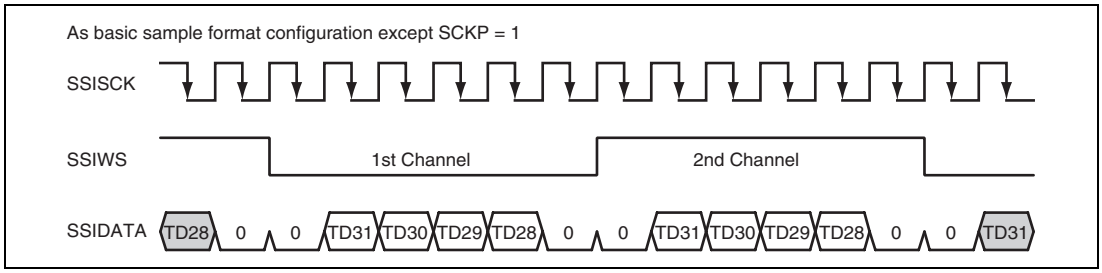


Figure 18.11 Inverted Clock

2. Inverted Word Select

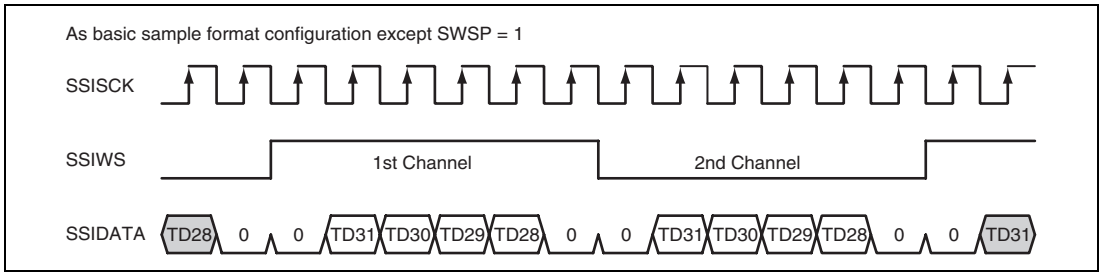


Figure 18.12 Inverted Word Select

3. Inverted Padding Polarity

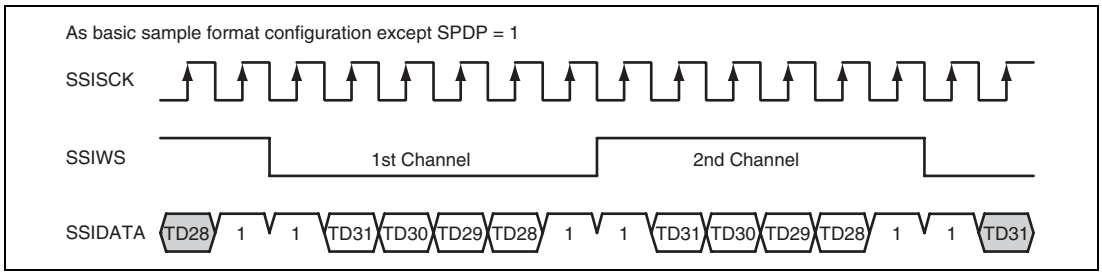


Figure 18.13 Inverted Padding Polarity

4. Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

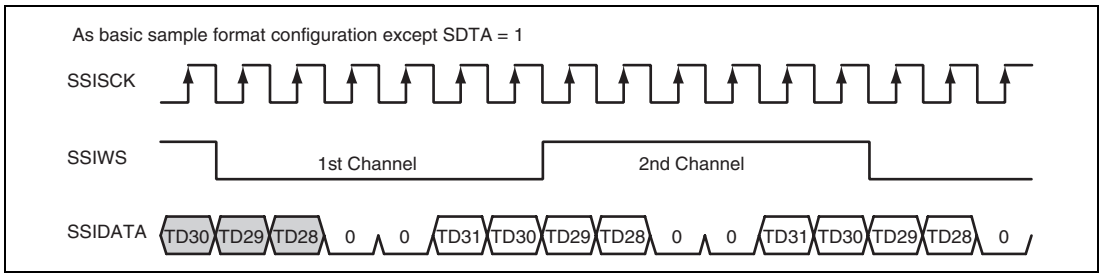


Figure 18.14 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

5. Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

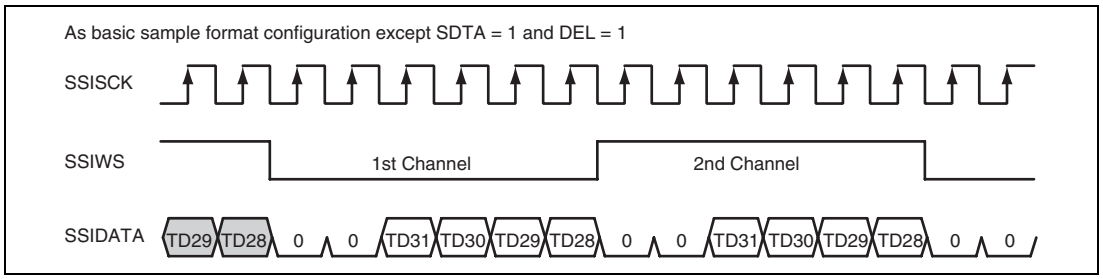


Figure 18.15 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

6. Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

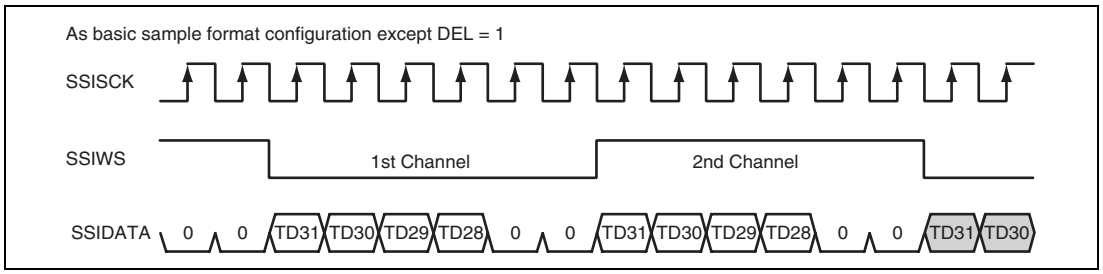


Figure 18.16 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

7. Parallel Right-Aligned with Delay

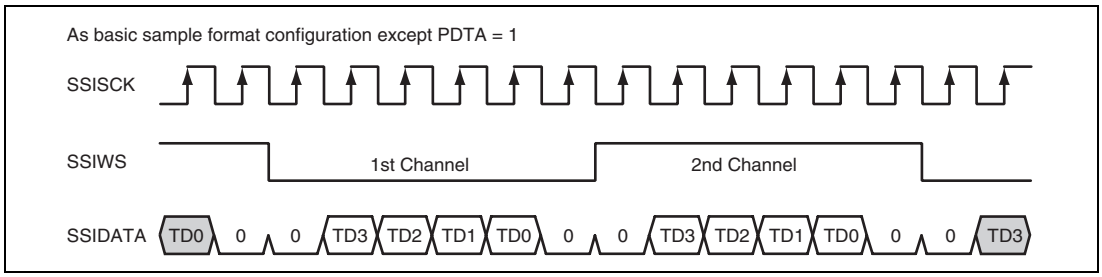


Figure 18.17 Parallel Right-Aligned with Delay

8. Mute Enabled

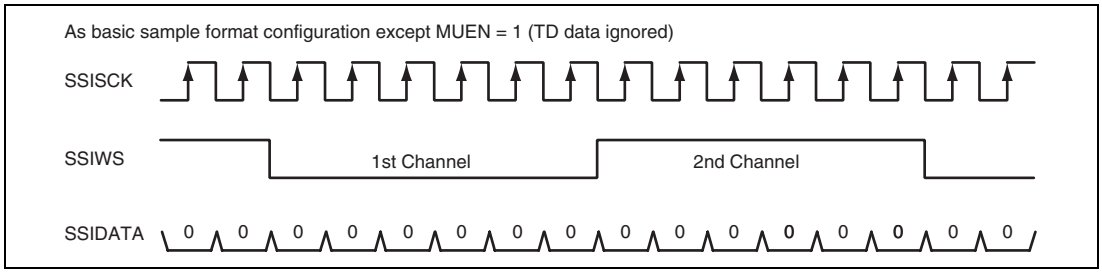


Figure 18.18 Mute Enabled

18.4.3 Operation Modes

There are three modes of operation: configuration, enabled and disabled. Figure 18.19 shows how the module enters each of these modes.

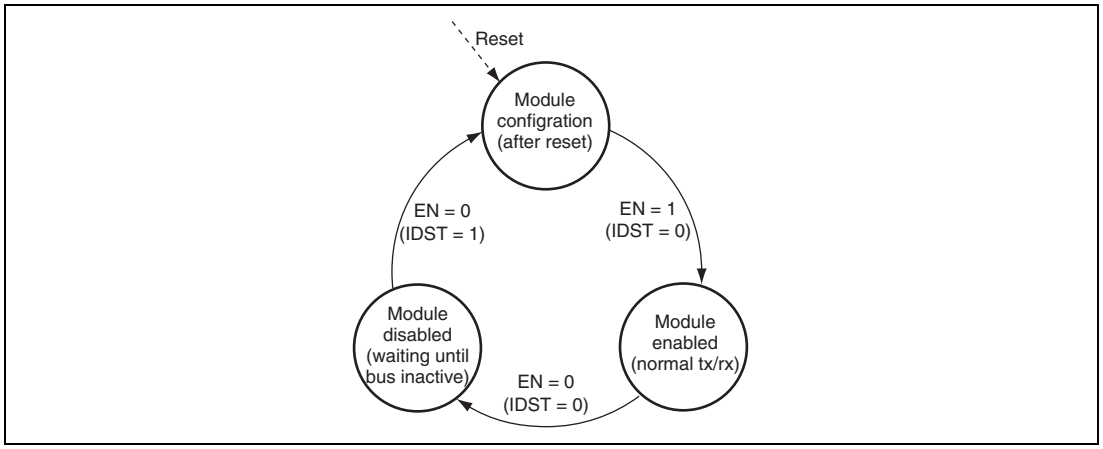


Figure 18.19 Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required configuration fields in the control register should be defined in this mode, before the SSI module is enabled by setting the EN bit.

Setting the EN bit causes the module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode is dependent on the operation mode selected. For details, refer to section 18.4.4, Transmit Operation and section 18.4.5, Receive Operation, below.

18.4.4 Transmit Operation

Transmission can be controlled either by DMA or interrupt.

DMA control is preferred to reduce the processor load. In DMA control mode the processor will only receive interrupts if there is an underflow or overflow of data or the DMAC has finished its transfer.

The alternative method is using the interrupts that the SSI module generates to supply data as required. This mode has a higher interrupt load as the module is only double buffered and will require data to be written at least every system word period.

When disabling the module, the SSI clock* must remain present until the SSI module is in idle state, indicated by the IIRQ bit.

Figure 18.20 shows the transmit operation in DMA control mode, and figure 18.21 shows the transmit operation in interrupt control mode.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Transmission Using DMA Controller

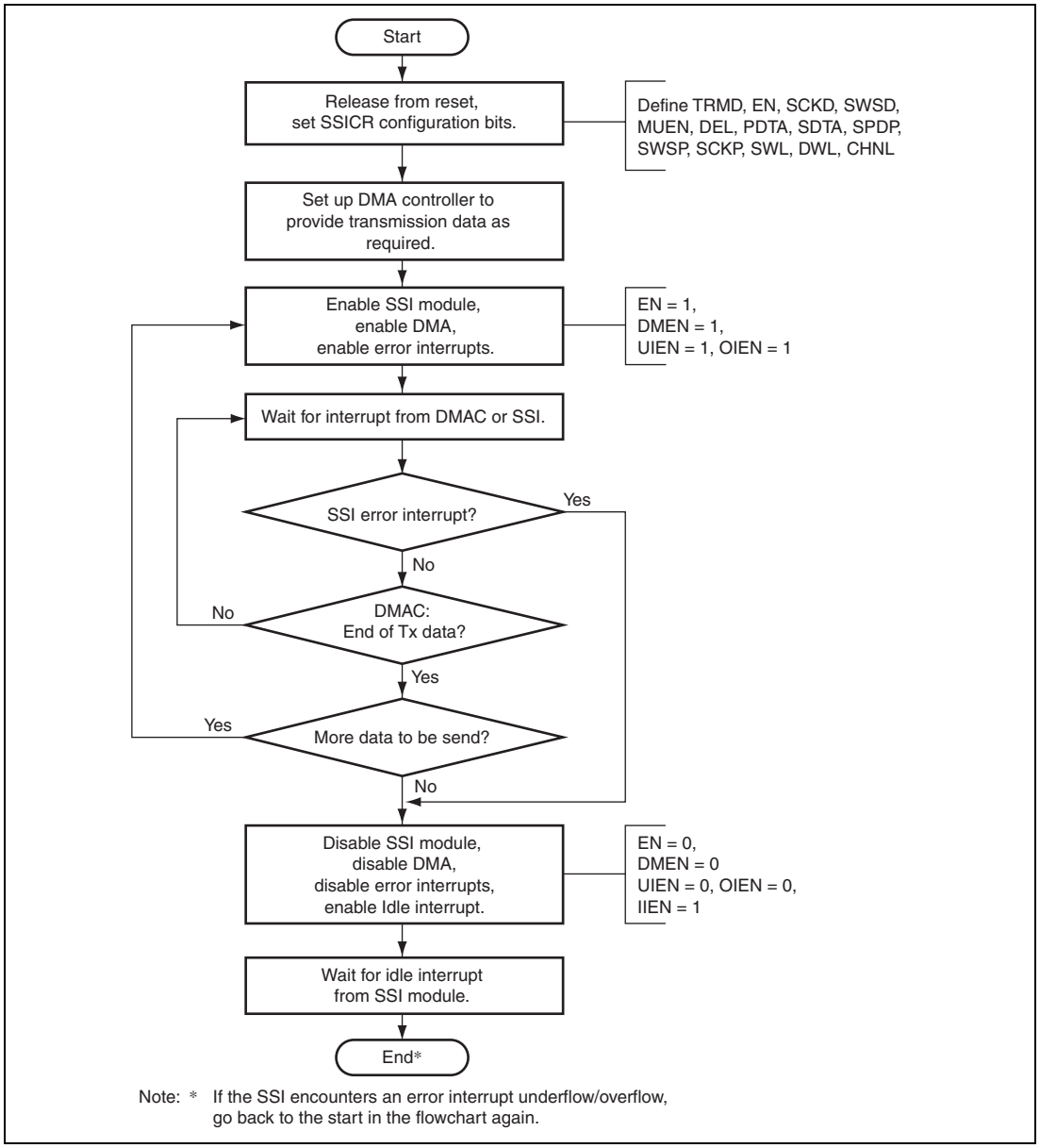


Figure 18.20 Transmission Using DMA Controller

(2) Transmission using Interrupt Data Flow Control

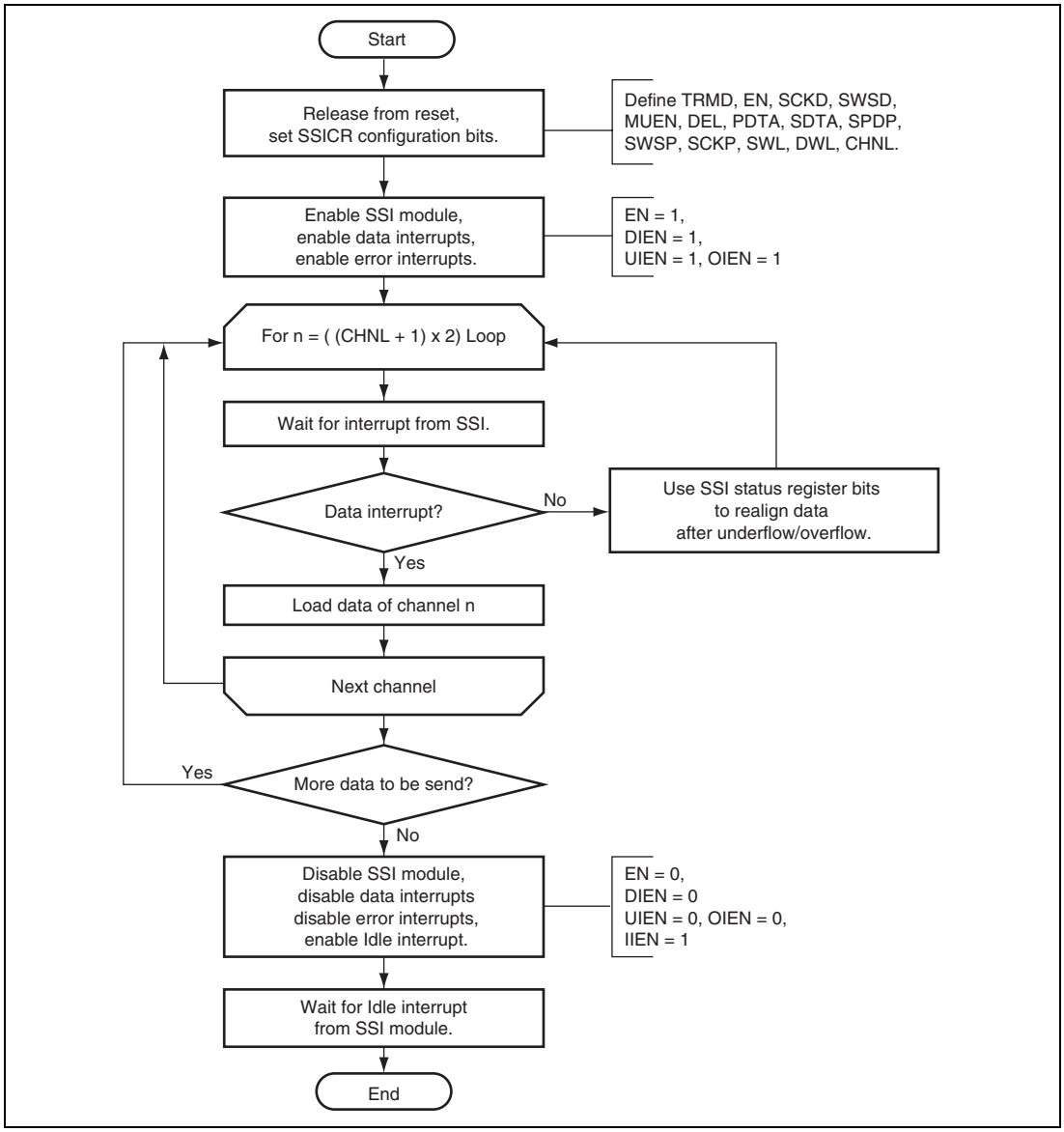


Figure 18.21 Transmission Using Interrupt Data Flow Control

18.4.5 Receive Operation

Like transmission, reception can be controlled either by DMA or interrupt.

Figures 18.22 and 18.23 show the flow of operation.

When disabling the SSI module, the SSI clock* must be kept supplied until the IIRQ bit is in idle state.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Reception Using DMA Controller

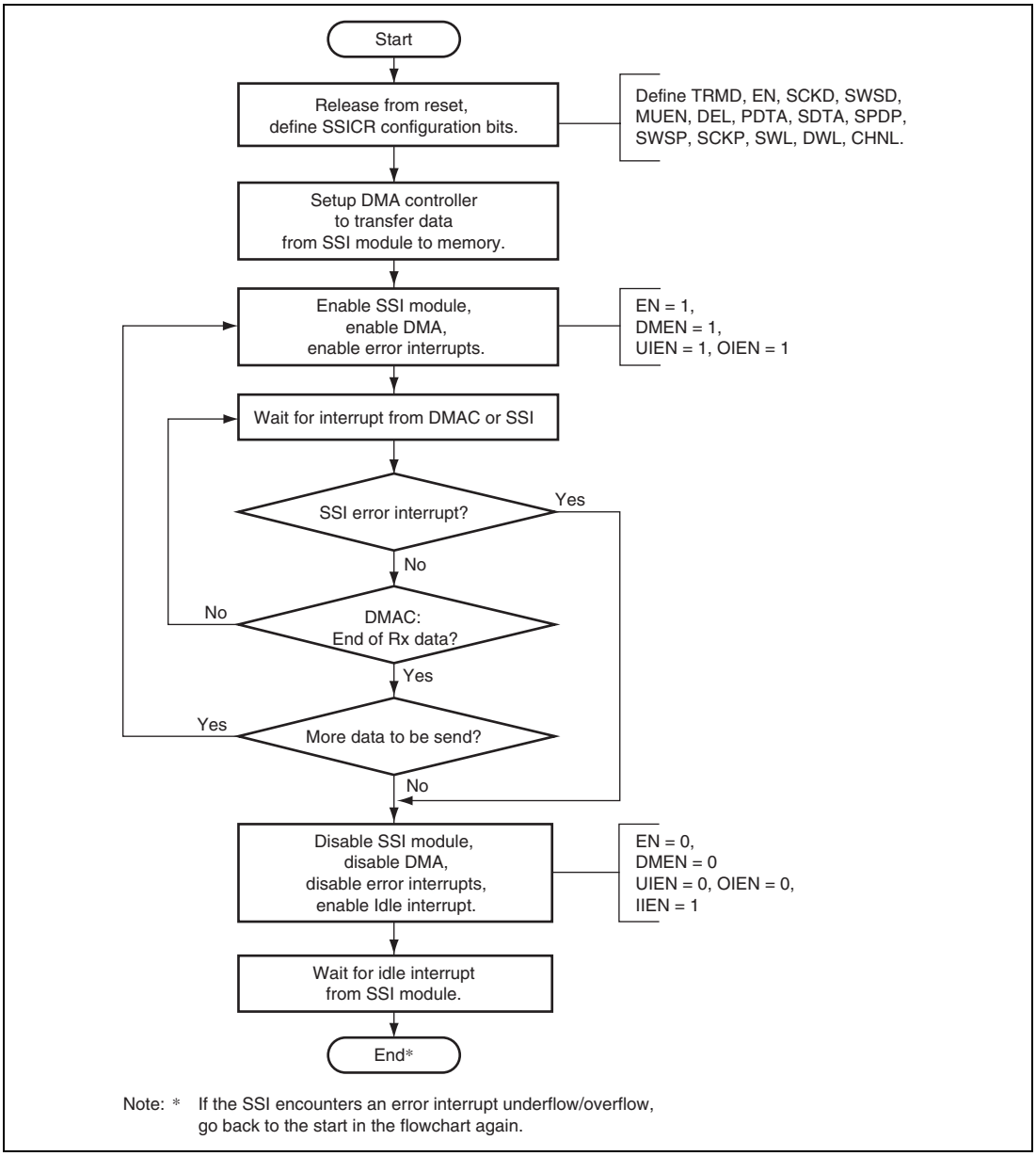


Figure 18.22 Reception Using DMA Controller

(2) Reception Using Interrupt Data Flow Control

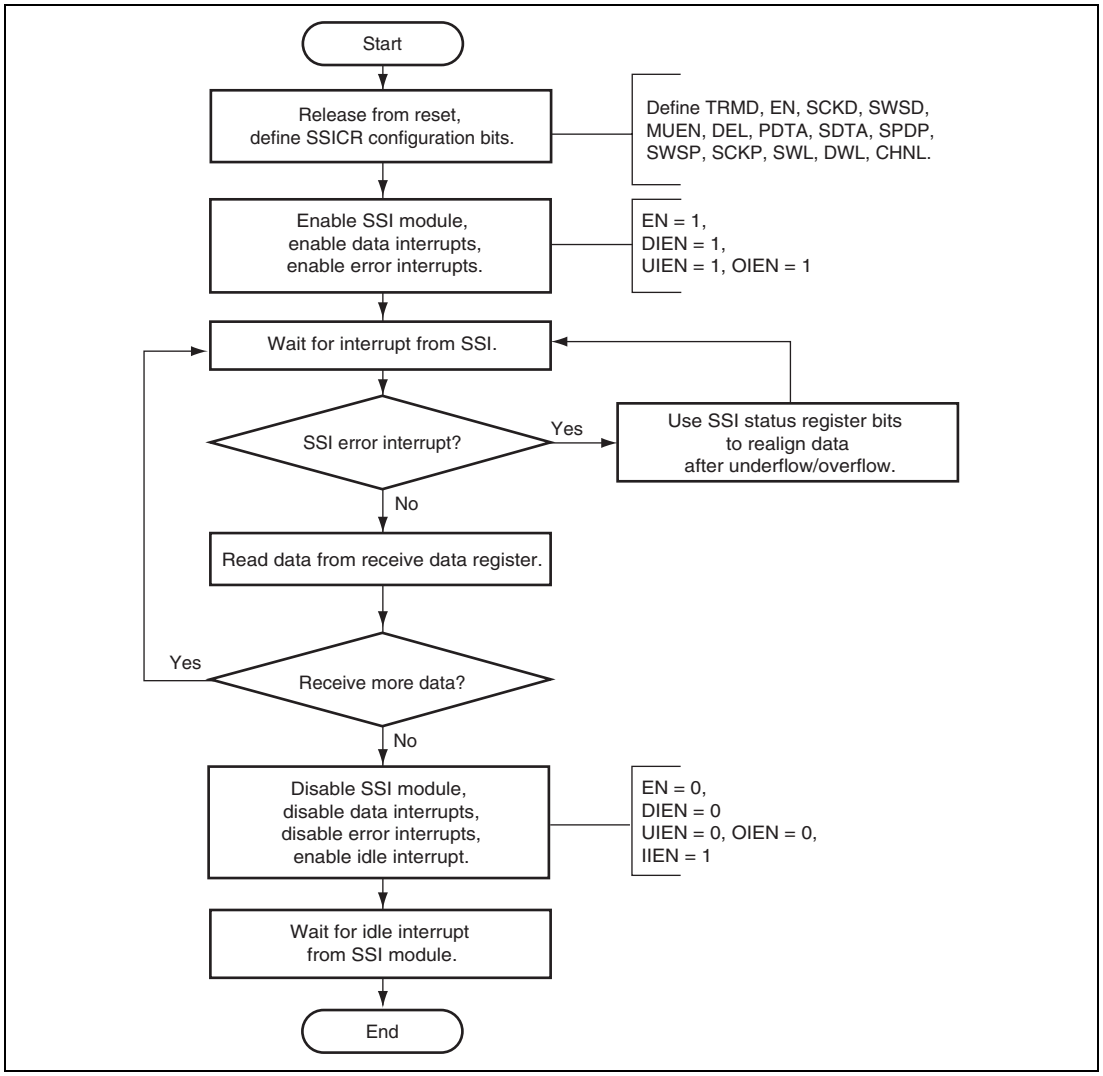


Figure 18.23 Reception Using Interrupt Data Flow Control

When an underflow or overflow error condition has matched, the CHNO [1:0] bit and the SWNO bit can be used to recover the SSI module to a known status. When an underflow or overflow occurs, the host can read the channel number and system word number to determine what point the serial audio stream has reached. In the transmitter case, the host can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSI module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case the host CPU can store null data to make the number of receive data items consistent until it is ready to store the sample data that the SSI module is indicating will be received next, and so resynchronize with the audio data stream.

18.4.6 Temporary Stop and Restart Procedures in Transmit Mode

The following procedures can be used for implementation.

(1) Procedure for the Repeated Transfer and Stop without having to Reconfigure the DMAC

1. Set SSICR.DMEN = 0 (disabling a DMA request) to stop the DMA transfer.
2. Wait for SSISR.DIRQ = 1 (transmit mode: the transmit buffer is empty) using a polling, interrupt, or the like.
3. With SSICR.EN = 0 (disabling an SSI module operation), stop the transfer.
4. Before attempting another transfer, make sure that SSISR.IDST = 1 is reached.
5. Set SSICR.EN = 1 (enabling an SSI module operation).
6. Wait for SSISR.DIRQ = 1, using a polling, interrupt, or the like.
7. Setting SSICR.DMEN = 1 (enabling a DMA request) will restart the DMA transfer.

(2) Procedure for Reconfiguring the DMAC after an SSI stop

1. Set SSICR.DMEN = 0 (disabling a DMA request) to stop the DMA transfer.
2. Wait for SSISR.DIRQ = 1 (transmit mode: the transmit buffer is empty), using a polling, interrupt, or the like.
3. With SSICR.EN = 0 (disabling an SSI module operation), stop the transfer.
4. Stop the DMAC with DMSCNT of the DMAC.
5. Before attempting another transfer, make sure that SSISR.IDST = 1 is reached.
6. Set SSICR.EN = 1 (enabling an SSI module operation).
7. Set the DMAC registers and start the transfer.
8. Setting SSICR.DMEN = 1 (enabling a DMA request) will restart the DMA transfer.

18.4.7 Serial Bit Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSI module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.

If the serial clock direction is set to output (SCKD = 1), this module is in clock master mode, and the shift register uses the oversampling clock or a divided oversampling clock as the bit clock. The oversampling clock is divided by the ratio specified by the serial oversampling clock division ratio bits (CKDV) in SSICR for use as the bit clock by the shift register.

In either case the module pin, SSISCK, is the same as the bit clock.

18.5 Usage Notes

18.5.1 Limitations from Overflow during Receive DMA Operation

If an overflow occurs while the receive DMA is in operation, the module should be restarted. The receive buffer in the SSI consists of 32-bit registers that share the L and R channels. Therefore, data to be received at the L channel may sometimes be received at the R channel if an overflow occurs, for example, under the following condition: the control register (SSICR) has a 32-bit setting for both data word length (DWL2 to DWL0) and system word length (SWL2 to SWL0).

If an overflow is confirmed with the overflow error interrupt or overflow error status flag (the OIRQ bit in SSISR), write 0 to the EN bit in SSICR and DMEN bit to disable DMA in the SSI module, thus stopping the operation. (In this case, the controller setting should also be stopped.) After this, write 0 to the OIRQ bit to clear the overflow status, set DMA again and restart the transfer.

18.5.2 Note on Using Oversample Clock

To use the externally input clock as the oversample clock, refer to the section 4.6.1, Note on Inputting External Clock, in which the terms EXTAL and XTAL pins should be replaced by the AUDIO_X1 and AUDIO_X2 pins respectively.

To use the crystal resonator, refer to the section 4.6.2, Note on Using Crystal Resonator, in which the terms EXTAL and XTAL pins should be replaced by the AUDIO_X1 and AUDIO_X2 pins respectively.

Also, see section 4.6.3, Note on Resonator.

18.5.3 Restriction on Stopping Clock Supply

Once the bits MSTP53 and MSTP52 in the standby control register 5 (STBCR5) are cleared to 0 and the SSI operation is started, do not set these bits to 1 (stops clock supply to the SSI).

Section 19 Controller Area Network (RCAN-ET)

19.1 Summary

19.1.1 Overview

This document primarily describes the programming interface for the RCAN-ET module. It serves to facilitate the hardware/software interface so that engineers involved in the RCAN-ET implementation can ensure the design is successful.

19.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the RCAN-ET module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

19.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the RCAN-ET user interface LSI engineers must use this document to understand the hardware requirements.

19.1.4 References

1. CAN License Specification, Robert Bosch GmbH, 1992
2. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
3. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
4. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
5. Road vehicles - Controller area network (CAN): Part 1: Data link layer and physical signalling (ISO-11898-1, 2002)

19.1.5 Features

- Supports CAN specification 2.0B
- Bit timing compliant with ISO-11898-1
- 16 Mailbox version
- Clock 16 to 40 MHz
- 15 programmable Mailboxes for transmit/receive + 1 receive-only mailbox
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- Programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- Programmable CAN data rate up to 1MBit/s
- Transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- Data buffer access without SW handshake requirement in reception
- Flexible micro-controller interface
- Flexible interrupt structure

19.2 Architecture

19.2.1 Block Diagram

The RCAN-ET device offers a flexible and sophisticated way to organise and control CAN frames, providing the compliance to CAN2.0B Active and ISO-11898-1. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control and CAN Interface. The figure below shows the block diagram of the RCAN-ET Module. The bus interface timing is designed according to the peripheral bus I/F required for each product.

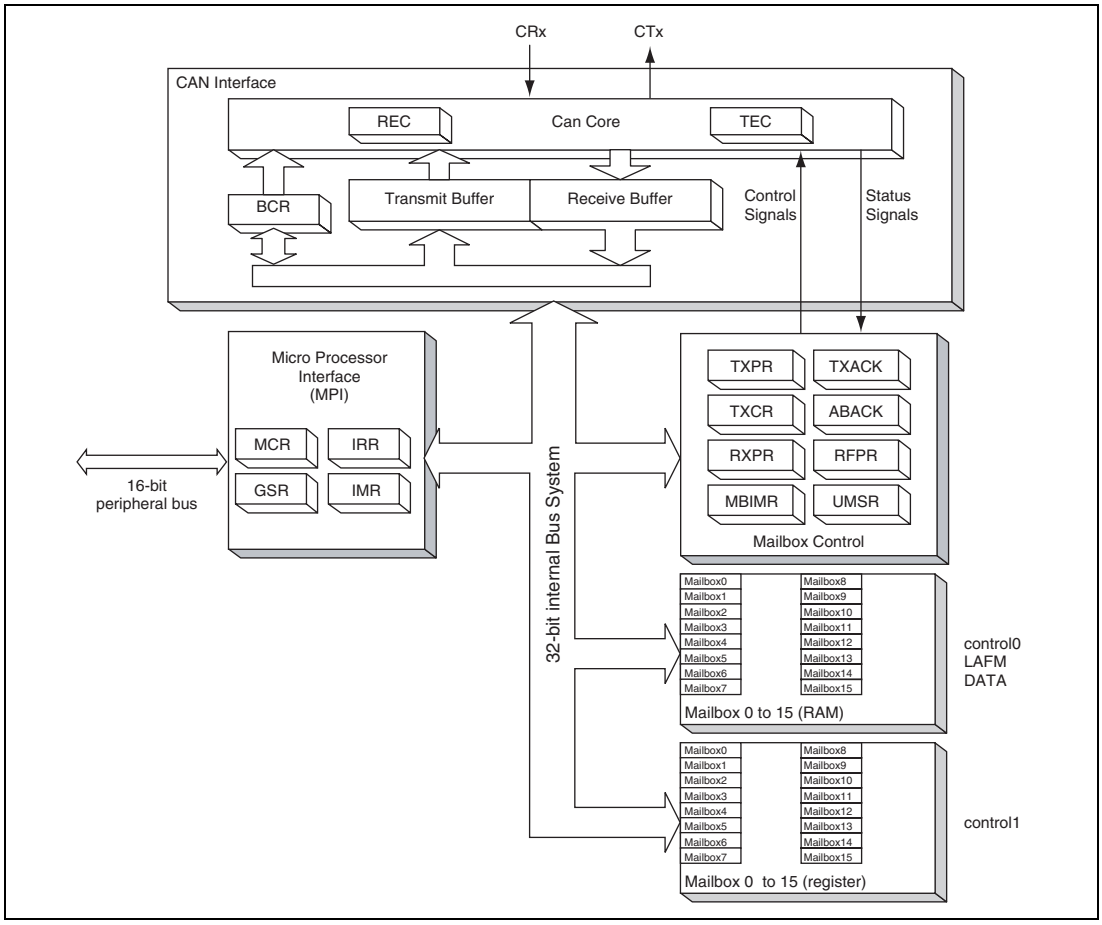


Figure 19.1 RCAN-ET Architecture

Important: Although core of RCAN-ET is designed based on a 32-bit bus system, the whole RCAN-ET including MPI for the CPU has 16-bit bus interface to CPU. LongWord (32-bit) accesses are converted into two consecutive word accesses by the bus interface.

19.2.2 Functions of Each Block

(1) Micro Processor Interface (MPI)

The MPI allows communication between the Renesas CPU and RCAN-ET's registers/mailboxes to control the memory interface. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of RCAN-ET so that the RCAN-ET can automatically exit the Sleep mode.

It contains registers such as MCR, IRR, GSR and IMR.

(2) Mailbox

The Mailboxes consists of RAM configured as message buffers and registers. There are 16 Mailboxes, and each mailbox has the following information.

<RAM>

- CAN message control (identifier, rtr, ide,etc)
- CAN message data (for CAN Data frames)
- Local Acceptance Filter Mask for reception

<Registers>

- CAN message control (dlc)
- 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit

(3) Mailbox Control

The Mailbox Control handles the following functions:

- For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.
- To transmit messages, RCAN-ET will run the internal arbitration to pick the correct priority message, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly.
- Arbitrates Mailbox accesses between the CPU and the Mailbox Control.
- Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, UMSR and MBIMR.

(4) CAN Interface

This block conforms to the requirements for a CAN Bus Data Link Controller which is specified in Ref. [3, 5]. It fulfils all the functions of a standard Data Link Controller as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

19.2.3 Input/Output Pins

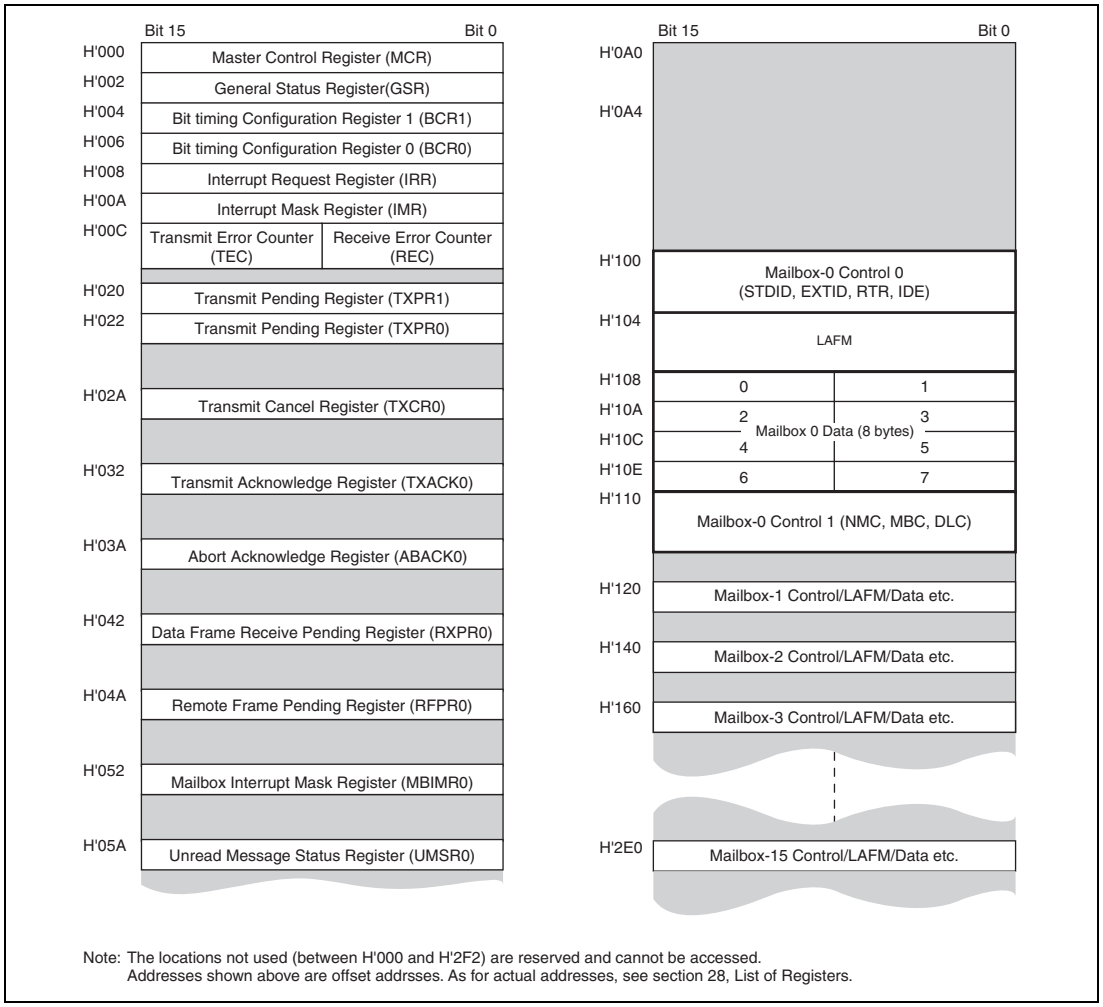
Table 19.1 shows the pin configuration of the RCAN-ET.

Table 19.1 Pin Configuration

Channel	Name	Abbreviation	I/O	Function
0	Transmit data pin	CTx0	Output	CAN-bus transmit pin
	Receive data pin	CRx0	Input	CAN-bus receive pin
1	Transmit data pin	CTx1	Output	CAN-bus transmit pin
	Receive data pin	CRx1	Input	CAN-bus receive pin

19.2.4 Memory Map

The diagram of the memory map is shown below.



Note: The locations not used (between H'000 and H'2F2) are reserved and cannot be accessed.
Addresses shown above are offset addrsses. As for actual addresses, see section 28, List of Registers.

Figure 19.2 RCAN-ET Memory Map

19.3 Mailbox

19.3.1 Mailbox Structure

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of 3 identical storage fields that are 1): Message Control, 2): Local Acceptance Filter Mask, 3): Message Data. The following table shows the address map for the control, LAFM, data and addresses for each mailbox.

Table 19.2 Address Map for Each Mailbox

	Address			
	Control0	**LAFM**	**Data**	**Control1**
Mailbox	**4 bytes**	**4 bytes**	**8 bytes**	**2 bytes**
0 (Receive Only)	H'100 to H'103	H'104 to H'107	H'108 to H'10F	H'110 to H'111
1	H'120 to H'123	H'124 to H'127	H'128 to H'12F	H'130 to H'131
2	H'140 to H'143	H'144 to H'147	H'148 to H'14F	H'150 to H'151
3	H'160 to H'163	H'164 to H'167	H'168 to H'16F	H'170 to H'171
4	H'180 to H'183	H'184 to H'187	H'188 to H'18F	H'190 to H'191
5	H'1A0 to H'1A3	H'1A4 to H'1A7	H'1A8 to H'1AF	H'1B0 to H'1B1
6	H'1C0 to H'1C3	H'1C4 to H'1C7	H'1C8 to H'1CF	H'1D0 to H'1D1
7	H'1E0 to H'1E3	H'1E4 to H'1E7	H'1E8 to H'1EF	H'1F0 to H'1F1
8	H'200 to H'203	H'204 to H'207	H'208 to H'20F	H'210 to H'211
9	H'220 to H'223	H'224 to H'227	H'228 to H'22F	H'230 to H'231
10	H'240 to H'243	H'244 to H'247	H'248 to H'24F	H'250 to H'251
11	H'260 to H'263	H'264 to H'267	H'268 to H'26F	H'270 to H'271
12	H'280 to H'283	H'284 to H'287	H'288 to H'28F	H'290 to H'291
13	H'2A0 to H'2A3	H'2A4 to H'2A7	H'2A8 to H'2AF	H'2B0 to H'2B1
14	H'2C0 to H'2C3	H'2C4 to H'2C7	H'2C8 to H'2CF	H'2D0 to H'2D1
15	H'2E0 to H'2E3	H'2E4 to H'2E7	H'2E8 to H'2EF	H'2F0 to H'2F1

Mailbox-0 is a receive-only box, and all the other Mailboxes can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

Table 19.3 Roles of Mailboxes

	Tx	Rx
MB15 to MB1	OK	OK
MB0	—	OK

MB0 (reception MB)

Byte: 8-bit access, Word: 16-bit access, LW (LongWord): 32-bit access

Regiter Name	Address	Data Bus																Access Size	Field Name
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MB[0].CONTROL0H	H'100	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
MB[0].CONTROL0L	H'102	EXTID[15:0]																Word	
MB[0].LAFMH	H'104	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
MB[0].LAFML	H'106	EXTID_LAFM[15:0]																Word	
MB[0].MSG_DATA[0][1]	H'108	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
MB[0].MSG_DATA[2][3]	H'10A	MSG_DATA_2								MSG_DATA_3								Byte/Word	
MB[0].MSG_DATA[4][5]	H'10C	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
MB[0].MSG_DATA[6][7]	H'10E	MSG_DATA_6								MSG_DATA_7								Byte/Word	
MB[0].CONTROL1H, L	H'110	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

MB1 to 15 (MB for transmission/reception)

Register Name	Address	Data Bus																Access Size	Field Name
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MB[n].CONTROL0H	H'100 + n × 32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
MB[n].CONTROL0L	H'102 + n × 32	EXTID[15:0]																Word	
MB[n].LAFMH	H'104 + n × 32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		Word/LW	LAFM
MB[n].LAFML	H'106 + n × 32	EXTID_LAFM[15:0]																Word	
MB[n].MSG_DATA[0][1]	H'108 + n × 32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte/Word/LW	Data
MB[n].MSG_DATA[2][3]	H'10A + n × 32	MSG_DATA_2								MSG_DATA_3								Byte/Word	
MB[n].MSG_DATA[4][5]	H'10C + n × 32	MSG_DATA_4								MSG_DATA_5								Byte/Word/LW	
MB[n].MSG_DATA[6][7]	H'10E + n × 32	MSG_DATA_6								MSG_DATA_7								Byte/Word	
MB[n].CONTROL1H, L	H'110 + n × 32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				Byte/Word	Control 1

Notes: 1. All bits shadowed in grey are reserved and the write value should be 0. The value returned by a read may not always be 0 and should not be relied upon.
2. MBC1 bit in mailbox is fixed to 1.
3. ATX and DART are not supported by mailbox-0, and the MBC setting of mailbox-0 is limited.
4. When the MCR15 bit is 1, the order of STDID, RTR, IDE and EXTID of both message control and LAFM differs from HCAN2.
5. n = 0 to 15 (mailbox number)

Figure 19.3 Mailbox-n Structure

19.3.2 Message Control Field

STDID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXTID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC = B'001, the RTR bit will never be set. When a Remote Frame is received, the CPU can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt), however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged. In case of overrun condition, the message received is discarded. Consequently, when a remote frame is causing overrun (UMSR is set) into a Mailbox configured with ATX = 1/NMC = 0, the transmission of the corresponding data frame is not carried out.

Important: In order to support automatic answer to remote frame when MBC = B'001 is used and ATX = 1 the RTR flag must be programmed to zero to allow data frame to be transmitted.

Note: when a Mailbox is configured to send a remote frame request the DLC used for transmission is the one stored into the Mailbox.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit): Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

- Mailbox-0

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

Note: MBC[1] of MB0 is always "1".

- Mailbox-15 to 1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]			
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

NMC (New Message Control): When this bit is set to '0', the Mailbox of which the RXPR or RFPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR or RFPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.

Important: Please note that if a remote frame is overwritten with a data frame or vice versa could be that both RXPR and RFPR flags (together with UMSR) are set for the same Mailbox. In this case the RTR bit within the Mailbox Control Field should be relied upon.

NMC	Description
0	Overrun mode (Initial value)
1	Overwrite mode

ATX (Automatic Transmission of Data Frame): When this bit is set to '1' and a Remote Frame is received into the Mailbox DLC is stored. Then, a Data Frame is transmitted from the same Mailbox using the current contents of the message data and updated DLC by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by ID priority or Mailbox priority as configured with the Message Transmission Priority control bit (MCR.2). In order to use this function, MBC[2:0] needs to be programmed to be B'001. When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received. Application needs to guarantee that the DLC of the remote frame correspond to the DLC of the data frame requested.

Important: When ATX is used and MBC = B'001 the filter for the IDE bit cannot be used since ID of remote frame has to be exactly the same as that of data frame as the reply message.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the CPU will be notified by the corresponding RFPR set, however, as RCAN-ET needs to transmit the current message as a Data Frame, the RTR bit remains unchanged.

ATX	Description
0	Automatic Transmission of Data Frame disabled (Initial value)
1	Automatic Transmission of Data Frame enabled

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', RCAN-ET tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

DART	Description
0	Re-transmission enabled (Initial value)
1	Re-Transmission disabled

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC = B'111, the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. The MBC = B'110, B'101 and B'100 settings are prohibited. When the MBC is set to any other value, the LAFM field becomes available. Please don't set TXPR when MBC is set as reception. Similarly, please don't set TXPR, when MBC is set as remote frame transmission and RTR in Mailbox is cleared. There is no hardware protection, and TXPR remains set. MBC[1] of Mailbox-0 is fixed to "1" by hardware. This is to ensure that MB0 cannot be configured to transmit Messages.

Table 19.4 Mailbox Function Setting

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX* • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	Setting prohibited				
1	0	1	Setting prohibited				
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive (Initial value)				

Note: * In order to support automatic retransmission, RTR shall be "0" when MBC = B'001 and ATX = 1.
When ATX = 1 is used the filter for IDE must not be used

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0,1, 2, … 8 that will be transmitted in a data frame. Please note that when a remote frame request is transmitted the DLC value to be used must be the same as the DLC of the data frame that is requested.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 byte (Initial value)
0	0	0	1	Data Length = 1 byte
0	0	1	0	Data Length = 2 bytes
0	0	1	1	Data Length = 3 bytes
0	1	0	0	Data Length = 4 bytes
0	1	0	1	Data Length = 5 bytes
0	1	1	0	Data Length = 6 bytes
0	1	1	1	Data Length = 7 bytes
1	x	x	x	Data Length = 8 bytes

19.3.3 Local Acceptance Filter Mask (LAFM)

This area is used as Local Acceptance Filter Mask (LAFM) for receive boxes**.**

LAFM: When MBC is set to B'001, B'010, B'011, this field is used as LAFM Field. The LAFM is comprised of two 16-bit read/write areas as follows. It allows a Mailbox to accept more than one identifier.

Register Name	Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Acces Size	Feld Name
MB[n].LAFMH	H'104 + n × 32	IDE_ LAFM	0	0	STDID_LAFM[10:0]											EXTID_ LAFM[17:16]		Word/LW	LAFM Field
MB[n].LAFML	H'106 + n × 32	EXTID_LAFM[15:0]																Word	

Note: n = 0 to 15 (mailbox number)

Figure 19.4 Acceptance Filter

If a bit is set in the LAFM, then the corresponding bit of a received CAN identifier is ignored when the RCAN-ET searches a Mailbox with the matching CAN identifier. If the bit is cleared, then the corresponding bit of a received CAN identifier must match to the STDID/IDE/EXTID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

Important: RCAN-ET starts to find a matching identifier from Mailbox-15 down to Mailbox-0. As soon as RCAN-ET finds one matching, it stops the search. The message will be stored or not depending on the NMC and RXPR/RFPR flags. This means that, even using LAFM, a received message can only be stored into 1 Mailbox.

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STDID, RTR, IDE and EXTID may differ to the ones originally set as they are updated with the STDID, RTR, IDE and EXTID of the received message.

STD_LAFM[10:0] — Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding STD_ID bit is cared
1	Corresponding STD_ID bit is "don't cared"

EXT_LAFM[17:0] — Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding EXT_ID bit is cared
1	Corresponding EXT_ID bit is "don't cared"

IDE_LAFM — Filter mask bit for the CAN IDE bit.

IDE_LAFM	Description
0	Corresponding IDE_ID bit is cared
1	Corresponding IDE_ID bit is "don't cared"

19.3.4 Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the CAN bus is bit 7 through to bit 0.

19.4 RCAN-ET Control Registers

The following sections describe RCAN-ET control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Table 19.5 RCAN-ET Control Registers Configuration

Description	Address	Name	Access Size (bits)
Master Control Register	000	MCR	Word
General Status Register	002	GSR	Word
Baud Rate Configuration Register 1	004	BCR1	Word
Baud Rate Configuration Register 0	006	BCR0	Word
Interrupt Request Register	008	IRR	Word
Interrupt Mask Register	00A	IMR	Word
Error Counter Register	00C	TEC/REC	Word

19.4.1 Master Control Register (MCR)

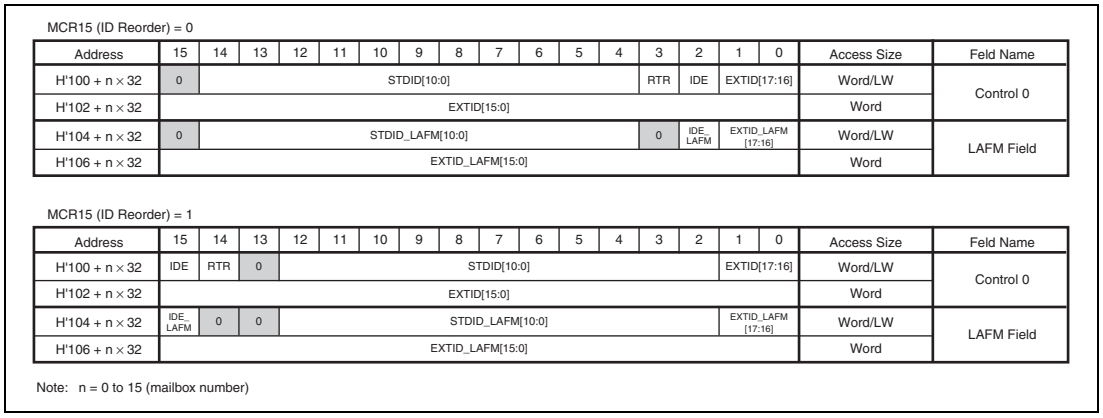
The Master Control Register (MCR) is a 16-bit read/write register that controls RCAN-ET.

- MCR (Address = H'000)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	—	—	—	TST[2:0]			MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

Bit 15 — ID Reorder (MCR15): This bit changes the order of STDID, RTR, IDE and EXTID of both message control and LAFM.

Bit15: MCR15	Description
0	RCAN-ET is the same as HCAN2
1	RCAN-ET is not the same as HCAN2 (Initial value)



MCR15 (ID Reorder) = 0

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Feld Name
H'100 + n × 32	0	STDID[10:0]											RTR	IDE	EXTID[17:16]		Word/LW	Control 0
H'102 + n × 32	EXTID[15:0]																Word	
H'104 + n × 32	0	STDID_LAFM[10:0]											0	IDE_LAFM	EXTID_LAFM [17:16]		Word/LW	LAFM Field
H'106 + n × 32	EXTID_LAFM[15:0]																Word	

MCR15 (ID Reorder) = 1

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Access Size	Feld Name
H'100 + n × 32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		Word/LW	Control 0
H'102 + n × 32	EXTID[15:0]																Word	
H'104 + n × 32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM [17:16]		Word/LW	LAFM Field
H'106 + n × 32	EXTID_LAFM[15:0]																Word	

Note: n = 0 to 15 (mailbox number)

Figure 19.5 ID Reorder

This bit can be modified only in reset mode.

Bit 14 — Auto Halt Bus Off (MCR14): If both this bit and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

Bit14: MCR14	Description
0	RCAN-ET remains in BusOff for normal recovery sequence (128 × 11 Recessive Bits) (Initial value)
1	RCAN-ET moves directly into Halt Mode after it enters BusOff if MCR6 is set.

This bit can be modified only in reset mode.

Bit 13 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 12 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 11 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 10 - 8 — Test Mode (TST[2:0]): This bit enables/disables the test modes. Please note that before activating the Test Mode it is requested to move RCAN-ET into Halt mode or Reset mode. This is to avoid that the transition to Test Mode could affect a transmission/reception in progress. For details, please refer to section 19.6.2, Test Mode Settings.

Please note that the test modes are allowed only for diagnosis and tests and not when RCAN-ET is used in normal operation.

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal mode (initial value)
0	0	1	Listen-only mode (receive-only mode)
0	1	0	Self test mode 1 (external)
0	1	1	Self test mode 2 (internal)
1	0	0	Write error counter
1	0	1	Error passive mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Bit 7 — Auto-wake Mode (MCR7): MCR7 enables or disables the Auto-wake mode. If this bit is set, the RCAN-ET automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the RCAN-ET does not automatically cancel the sleep mode.

RCAN-ET cannot store the message that wakes it up.

Note: MCR7 cannot be modified while in sleep mode.

Bit7: MCR7	Description
0	Auto-wake by CAN bus activity disabled (Initial value)
1	Auto-wake by CAN bus activity enabled

Bit 6 — Halt during Bus Off (MCR6): MCR6 enables or disables entering Halt mode immediately when MCR1 is set during Bus Off. This bit can be modified only in Reset or Halt mode. Please note that when Halt is entered in Bus Off the CAN engine is also recovering immediately to Error Active mode.

Bit6: MCR6	Description
0	Don't enter Halt mode during Bus Off but wait up to end of recovery sequence (Initial value)
1	Enter Halt mode immediately during Bus Off if MCR[1] or MCR[14] are asserted.

Bit 5 — Sleep Mode (MCR5): Enables or disables Sleep mode transition. If this bit is set, while RCAN-ET is in halt mode, the transition to sleep mode is enabled. Setting MCR5 is allowed after entering Halt mode. The two Error Counters (REC, TEC) will remain the same during Sleep mode. This mode will be exited in two ways:

1. by writing a '0' to this bit position,
2. or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus.

If Auto wake up mode is disabled, RCAN-ET will ignore all CAN bus activities until the sleep mode is terminated. When leaving this mode the RCAN-ET will synchronise to the CAN bus (by checking for 11 recessive bits) before joining CAN Bus activity. This means that, when the No.2 method is used, RCAN-ET will miss the first message to receive. CAN transceivers stand-by mode will also be unable to cope with the first message when exiting stand by mode, and the S/W needs to be designed in this manner.

In sleep mode only the following registers can be accessed: MCR, GSR, IRR and IMR.

Important: RCAN-ET is required to be in Halt mode before requesting to enter in Sleep mode. That allows the CPU to clear all pending interrupts before entering sleep mode. Once all interrupts are cleared RCAN-ET must leave the Halt mode and enter Sleep mode simultaneously (by writing MCR[5] = 1 and MCR[1] = 0 at the same time).

Bit 5: MCR5	Description
0	RCAN-ET sleep mode released (Initial value)
1	Transition to RCAN-ET sleep mode enabled

Bit 4 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 3 — Reserved. The written value should always be '0' and the returned value is '0'.

Bit 2 — Message Transmission Priority (MCR2): MCR2 selects the order of transmission for pending transmit data. If this bit is set, pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-15 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission).

If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field (STDID + IDE bit + EXTID (if IDE = 1) + RTR bit) with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit (internal arbitration works in the same way as the arbitration on the CAN Bus between two CAN nodes starting transmission at the same time).

This bit can be modified only in Reset or Halt mode.

Bit 2: MCR2	Description
0	Transmission order determined by message identifier priority (Initial value)
1	Transmission order determined by mailbox number priority (Mailbox-15 → Mailbox-1)

Bit 1—Halt Request (MCR1): Setting the MCR1 bit causes the CAN controller to complete its current operation and then enter Halt mode (where it is cut off from the CAN bus). The RCAN-ET remains in Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity and does not store messages or transmit messages. All the user registers (including Mailbox contents and TEC/REC) remain unchanged with the exception of IRR0 and GSR4 which are used to notify the halt status itself. If the CAN bus is in idle or intermission state regardless of MCR6, RCAN-ET will enter Halt Mode within one Bit Time. If MCR6 is set, a halt request during Bus Off will be also processed within one Bit Time. Otherwise the full Bus Off recovery sequence will be performed beforehand. Entering the Halt Mode can be notified by IRR0 and GSR4.

If both MCR14 and MCR6 are set, MCR1 is automatically set as soon as RCAN-ET enters BusOff.

In the Halt mode, the RCAN-ET configuration can be modified with the exception of the Bit Timing setting, as it does not join the bus activity. MCR[1] has to be cleared by writing a '0' in order to re-join the CAN bus. After this bit has been cleared, RCAN-ET waits until it detects 11 recessive bits, and then joins the CAN bus.

Note: After issuing a Halt request the CPU is not allowed to set TXPR or TXCR or clear MCR1 until the transition to Halt mode is completed (notified by IRR0 and GSR4). After MCR1 is set this can be cleared only after entering Halt mode or through a reset operation (SW or HW).

Note: Transition into or recovery from Halt mode, is only possible if the BCR1 and BCR0 registers are configured to a proper Baud Rate.

Bit 1: MCR1	Description
0	Clear Halt request (Initial value)
1	Halt mode transition request

Bit 0 — Reset Request (MCR0): Controls resetting of the RCAN-ET module. When this bit is changed from '0' to '1' the RCAN-ET controller enters its reset routine, re-initialising the internal logic, which then sets GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all user registers are initialised.

RCAN-ET can be re-configured while this bit is set. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, the RCAN-ET module waits until it detects 11 recessive bits, and then joins the CAN bus. The Baud Rate needs to be set up to a proper value in order to sample the value on the CAN Bus.

After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and RCAN-ET needs to be configured.

The Reset Request is equivalent to a Power On Reset but controlled by Software.

Bit 0: MCR0	Description
0	Clear Reset Request
1	CAN Interface reset mode transition request (Initial value)

19.4.2 General Status Register (GSR)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of RCAN-ET.

- GSR (Address = H'002)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bits 15 to 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 5 — Error Passive Status Bit (GSR5): Indicates whether the CAN Interface is in Error Passive or not. This bit will be set high as soon as the RCAN-ET enters the Error Passive state and is cleared when the module enters again the Error Active state (this means the GSR5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR5 and GSR0 must be considered.

Bit 5: GSR5	Description
0	RCAN-ET is not in Error Passive or in Bus Off status (Initial value) [Reset condition] RCAN-ET is in Error Active state
1	RCAN-ET is in Error Passive (if GSR0 = 0) or Bus Off (if GSR0 = 1) [Setting condition] When TEC ≥ 128 or REC ≥ 128 or if Error Passive Test Mode is selected

Bit 4 — Halt/Sleep Status Bit (GSR4): Indicates whether the CAN engine is in the halt/sleep state or not. Please note that the clearing time of this flag is not the same as the setting time of IRR12.

Please note that this flag reflects the status of the CAN engine and not of the full RCAN-ET IP. RCAN-ET exits sleep mode and can be accessed once MCR5 is cleared. The CAN engine exits sleep mode only after two additional transmission clocks on the CAN Bus.

Bit 4: GSR4	Description
0	RCAN-ET is not in the Halt state or Sleep state (Initial value)
1	Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1) [Setting condition] If MCR1 is set and the CAN bus is either in intermission or idle or MCR5 is set and RCAN-ET is in the halt mode or RCAN-ET is moving to Bus Off when MCR14 and MCR6 are both set

Bit 3 — Reset Status Bit (GSR3): Indicates whether the RCAN-ET is in the reset state or not.

Bit 3: GSR3	Description
0	RCAN-ET is not in the reset state
1	Reset state (Initial value) [Setting condition] After an RCAN-ET internal reset (due to SW or HW reset)

Bit 2 — Message Transmission in progress Flag (GSR2): Flag that indicates to the CPU if the RCAN-ET is in Bus Off or transmitting a message or an error/overload flag due to error detected during transmission. The timing to set TXACK is different from the time to clear GSR2. TXACK is set at the 7th bit of End Of Frame. GSR2 is set at the 3rd bit of intermission if there are no more messages ready to be transmitted. It is also set by arbitration lost, bus idle, reception, reset or halt transition.

Bit 2: GSR2	Description
0	RCAN-ET is in Bus Off or a transmission is in progress
1	[Setting condition] Not in Bus Off and no transmission in progress (Initial value)

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning.

Bit 1: GSR1	Description
0	[Reset condition] When (TEC < 96 and REC < 96) or Bus Off (Initial value)
1	[Setting condition] When 96 ≤ TEC < 256 or 96 ≤ REC < 256

Note: REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence. However the flag GSR1 is not set in Bus Off.

Bit 0—Bus Off Flag (GSR0): Flag that indicates that RCAN-ET is in the bus off state.

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state or after a HW or SW reset (Initial value)
1	[Setting condition] When TEC ≥ 256 (bus off state)

Note: Only the lower 8 bits of TEC are accessible from the user interface. The 9th bit is equivalent to GSR0.

19.4.3 Bit Configuration Register (BCR0, BCR1)

The bit configuration registers (BCR0 and BCR1) are 2 × 16-bit read/write register that are used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

The Time quanta is defined as:

$$\text{Timequanta} = \frac{2 \times \text{BRP}}{f_{clk}}$$

Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.

- BCR1 (Address = H'004)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				—	TSG2[2:0]			—	—	SJW[1:0]		—	—	—	BSP
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

Please refer to the table below for TSG1 and TSG2 setting.

Bits 15 to 12 — Time Segment 1 (TSG1[3:0] = BCR1[15:12]): These bits are used to set the segment TSEG1 (= PRSEG + PHSEG1) to compensate for edges on the CAN Bus with a positive phase error. A value from 4 to 16 time quanta can be set.

Bit 15: TSG1[3]	Bit 14: TSG1[2]	Bit 13: TSG1[1]	Bit 12: TSG1[0]	Description
0	0	0	0	Setting prohibited (Initial value)
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	PRSEG + PHSEG1 = 4 time quanta
0	1	0	0	PRSEG + PHSEG1 = 5 time quanta
:	:	:	:	:
:	:	:	:	:
1	1	1	1	PRSEG + PHSEG1 = 16 time quanta

Bit 11: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 10 to 8 — Time Segment 2 (TSG2[2:0] = BCR1[10:8]): These bits are used to set the segment TSEG2 (= PHSEG2) to compensate for edges on the CAN Bus with a negative phase error. A value from 2 to 8 time quanta can be set as shown below.

Bit 10: TSG2[2]	Bit 9: TSG2[1]	Bit 8: TSG2[0]	Description
0	0	0	Setting prohibited (Initial value)
0	0	1	PHSEG2 = 2 time quanta (conditionally prohibited) See the table below for TSG1 and TSG2 setting.
0	1	0	PHSEG2 = 3 time quanta
0	1	1	PHSEG2 = 4 time quanta
1	0	0	PHSEG2 = 5 time quanta
1	0	1	PHSEG2 = 6 time quanta
1	1	0	PHSEG2 = 7 time quanta
1	1	1	PHSEG2 = 8 time quanta

Bits 7 and 6: Reserved. The written value should always be '0' and the returned value is '0'.

Bits 5 and 4 - ReSynchronisation Jump Width (SJW[1:0] = BCR0[5:4]): These bits set the synchronisation jump width.

Bit 5: SJW[1]	Bit 4: SJW[0]	Description
0	0	Synchronisation Jump width = 1 time quantum (Initial value)
0	1	Synchronisation Jump width = 2 time quanta
1	0	Synchronisation Jump width = 3 time quanta
1	1	Synchronisation Jump width = 4 time quanta

Bits 3 to 1: Reserved. The written value should always be '0' and the returned value is '0'.

Bit 0 — Bit Sample Point (BSP = BCR1[0]): Sets the point at which data is sampled. Three-time sampling is only available when the BRP is programmed to be greater than 4.

Bit 0: BSP	Description
0	Bit sampling at one point (end of time segment 1) (Initial value)
1	Bit sampling at three points (rising edge of the last three clock cycles of PHSEG1)

- BCR0 (Address = H'006)

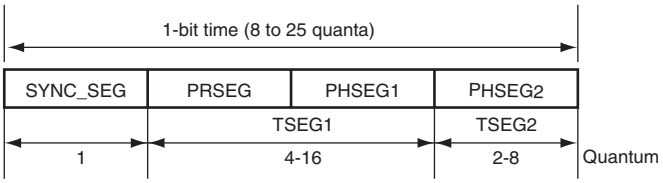
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BRP[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 8 to 15 : Reserved. The written value should always be '0' and the returned value is '0'.

Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral clock periods contained in a Time Quantum.

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 × peripheral clock (Initial value)
0	0	0	0	0	0	0	1	4 × peripheral clock
0	0	0	0	0	0	1	0	6 × peripheral clock
: :	: :	: :	: :	: :	: :	: :	: :	2 × (register value+1) × peripheral clock
1	1	1	1	1	1	1	1	512 × peripheral clock

- Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronisation of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronisation (resynchronisation) is established.)

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

TSEG1: TSG1 + 1

TSEG2: TSG2 + 1

The RCAN-ET Bit Rate Calculation is:

$$\text{Bit Rate} = \frac{f_{clk}}{2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1)}$$

where BRP is given by the register value and TSEG1 and TSEG2 are derived values from TSG1 and TSG2 register values.

f_{CLK} = Peripheral Clock

BCR Setting Constraints

$TSEG1min > TSEG2 \geq SJWmax$ (SJW = 1 to 4)

$8 \leq TSEG1 + TSEG2 + 1 \leq 25$ time quanta (TSEG1 + TSEG2 + 1 = 7 is not allowed)

$TSEG2 \geq 2$

These constraints allow the setting range shown in the table below for TSEG1 and TSEG2 in the Bit Configuration Register. The number in the table shows possible setting of SJW. "No" shows that there is no allowed combination of TSEG1 and TSEG2.

Table 19.6 TSG and TSEG Setting

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	No	1-3	No	No	No	No	No	
0100	5	1-2	1-3	1-4	No	No	No	No	
0101	6	1-2	1-3	1-4	1-4	No	No	No	
0110	7	1-2	1-3	1-4	1-4	1-4	No	No	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	No	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

Example 1: To have a Bit rate of 500 Kbps with a frequency of fclk = 40 MHz it is possible to set: BRP = 3, TSEG1 = 6, TSEG2 = 3.

Then the configuration to write is BCR1 = H'5200 and BCR0 = H'0003.

Example 2: To have a Bit rate of 250 Kbps with a frequency of fclk = 35 MHz it is possible to set: BRP = 4, TSEG1 = 8, TSEG2 = 5.

Then the configuration to write is BCR1 = H'7400 and BCR0 = H'0004.

19.4.4 Interrupt Request Register (IRR)

The interrupt request register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	IRR13	IRR12	—	—	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bits 15 to 14: Reserved.

Bit 13 - Message Error Interrupt (IRR13): This interrupt indicates that:

- A message error has occurred when in test mode.
- Note: If a Message Overload condition occurs when in Test Mode, then this bit will not be set. When not in test mode this interrupt is inactive.

Bit 13: IRR13	Description
0	message error has not occurred in test mode (Initial value) [Clearing condition] Writing 1
1	[Setting condition] message error has occurred in test mode

Bit 12 – Bus Activity while in Sleep Mode (IRR12): IRR12 indicates that a CAN bus activity is present. While the RCAN-ET is in sleep mode and a dominant bit is detected on the CAN bus, this bit is set. This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. If auto wakeup is not used and this interrupt is not requested it needs to be disabled by the related interrupt mask register. If auto wake up is not used and this interrupt is requested it should be cleared only after recovering from sleep mode. This is to avoid that a new falling edge of the reception line causes the interrupt to get set again.

Please note that the setting time of this interrupt is different from the clearing time of GSR4.

Bit 12: IRR12	Description
0	bus idle state (Initial value) [Clearing condition] Writing 1
1	CAN bus activity detected in RCAN-ET sleep mode [Setting condition] dominant bit level detection on the CRx line while in sleep mode

Bits 11 to 10: Reserved

Bit 9 – Message Overrun/Overwrite Interrupt Flag (IRR9): Flag indicating that a message has been received but the existing message in the matching Mailbox has not been read as the corresponding RXPR or RFPR is already set to '1' and not yet cleared by the CPU. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared when all bit in UMSR (Unread Message Status Register) are cleared (by writing '1') or by setting MBIMR (MailBox interrupt Mast Register) for all UMSR flag set . It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 9: IRR9	Description
0	No pending notification of message overrun/overwrite [Clearing condition] Clearing of all bit in UMSR/setting MBIMR for all UMSR set (initial value)
1	A receive message has been discarded due to overrun condition or a message has been overwritten [Setting condition] Message is received while the corresponding RXPR and/or RFPR = 1 and MBIMR = 0

Bit 8 - Mailbox Empty Interrupt Flag (IRR8): This bit is set when one of the messages set for transmission has been successfully sent (corresponding TXACK flag is set) or has been successfully aborted (corresponding ABACK flag is set). The related TXPR is also cleared and this mailbox is now ready to accept a new message data for the next transmission. In effect, this bit is set by an OR'ed signal of the TXACK and ABACK bits not masked by the corresponding MBIMR flag. Therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit position has no effect.

Bit 8: IRR8	Description
0	Messages set for transmission or transmission cancellation request NOT progressed. (Initial value) [Clearing Condition] All the TXACK and ABACK bits are cleared/setting MBIMR for all TXACK and ABACK set
1	Message has been transmitted or aborted, and new message can be stored [Setting condition] When one of the TXPR bits is cleared by completion of transmission or completion of transmission abort, i.e., when a TXACK or ABACK bit is set (if MBIMR = 0).

Bit 7 - Overload Frame (IRR7): Flag indicating that the RCAN-ET has detected a condition that should initiate the transmission of an overload frame. Note that on the condition of transmission being prevented, such as listen only mode, an Overload Frame will NOT be transmitted, but IRR7 will still be set. IRR7 remains asserted until reset by writing a '1' to this bit position - writing a '0' has no effect.

Bit 7: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	[Setting conditions] Overload condition detected

Bit 6 - Bus Off Interrupt Flag (IRR6): This bit is set when RCAN-ET enters the Bus-off state or when RCAN-ET leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of the Bus-off recovery sequence (128 × 11 consecutive recessive bits) or the transition from Bus Off to Halt (automatic or manual). This bit remains set even if the RCAN-ET node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR0 to judge whether RCAN-ET is in the bus-off or error active status. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect.

Bit 6: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Enter Bus off state caused by transmit error or Error Active state returning from Bus-off [Setting condition] When TEC becomes ≥ 256 or End of Bus-off after 128× 11 consecutive recessive bits or transition from Bus Off to Halt

Bit 5 - Error Passive Interrupt Flag (IRR5): Interrupt flag indicating the error passive state caused by the transmit or receive error counter or by Error Passive forced by test mode. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. Please note that the SW needs to check GSR0 and GSR5 to judge whether RCAN-ET is in Error Passive or Bus Off status.

Bit 5: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128 or Error Passive test mode is used

Bit 4 - Receive Error Counter Warning Interrupt Flag (IRR4): This bit becomes set if the receive error counter (REC) reaches a value greater than 95 when RCAN-ET is not in the Bus Off status. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 4: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96 and RCAN-ET is not in Bus Off

Bit 3 - Transmit Error Counter Warning Interrupt Flag (IRR3): This bit becomes set if the transmit error counter (TEC) reaches a value greater than 95. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect.

Bit 3: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 2 - Remote Frame Request Interrupt Flag (IRR2): Flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox, with related MBIMR not set, contains a remote frame transmission request. This bit is automatically cleared when all bits in the Remote Frame Receive Pending Register (RFPR), are cleared. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 2: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (Initial value)
1	at least one remote request is pending [Setting condition] When remote frame is received and the corresponding MBIMR = 0

Bit 1 – Data Frame Received Interrupt Flag (IRR1): IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Data Frame Receive Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR of the RXPR flags from each configured receive mailbox with related MBIMR not set. It is also cleared by writing a '1' to all the correspondent bit position in MBIMR. Writing to this bit has no effect.

Bit 1: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (Initial value)
1	Data frame received and stored in Mailbox [Setting condition] When data is received and the corresponding MBIMR = 0

Bit 0 – Reset/Halt/Sleep Interrupt Flag (IRR0): This flag can get set for three different reasons. It can indicate that:

1. Reset mode has been entered after a SW (MCR0) or HW reset
2. Halt mode has been entered after a Halt request (MCR1)
3. Sleep mode has been entered after a sleep request (MCR5) has been made while in Halt mode.

The GSR may be read after this bit is set to determine which state RCAN-ET is in.

Important: When a Sleep mode request needs to be made, the Halt mode must be used beforehand. Please refer to the MCR5 description and figure 19.8.

IRR0 is set by the transition from "0" to "1" of GSR3 or GSR4 or by transition from Halt mode to Sleep mode. So, IRR0 is not set if RCAN-ET enters Halt mode again right after exiting from Halt mode, without GSR4 being cleared. Similarly, IRR0 is not set by direct transition from Sleep mode to Halt Request. At the transition from Halt/Sleep mode to Transition/Reception, clearing GSR4 needs (one-bit time - TSEG2) to (one-bit time * 2 - TSEG2).

In the case of Reset mode, IRR0 is set, however, the interrupt to the CPU is not asserted since IMR0 is automatically set by initialisation.

Bit 0: IRR0	Description
0	[Clearing condition] Writing 1
1	Transition to S/W reset mode or transition to halt mode or transition to sleep mode (Initial value) [Setting condition] When reset/halt/sleep transition is completed after a reset (MCR0 or HW) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

19.4.5 Interrupt Mask Register (IMR)

The interrupt mask register is a 16 bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15 to 0: Maskable interrupt sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed.

Bit[15:0]: IMRn	Description
0	Corresponding IRR is not masked (IRQ is generated for interrupt conditions)
1	Corresponding interrupt of IRR is masked (Initial value)

19.4.6 Transmit Error Counter (TEC) and Receive Error Counter (REC)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [2], [3], [4] and [5]. When not in (Write Error Counter) test mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or entering to bus off.

In Write Error Counter test mode (i.e. TST[2:0] = B'100), it is possible to write to this register. The same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, RCAN-ET needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * It is only possible to write the value in test mode when TST[2:0] in MCR is B'100.
REC is incremented during Bus Off to count the recurrences of 11 recessive bits as requested by the Bus Off recovery sequence.

19.5 RCAN-ET Mailbox Registers

The following sections describe RCAN-ET Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: LongWord access is carried out as two consecutive Word accesses.

Table 19.7 RCAN-ET Mailbox Registers

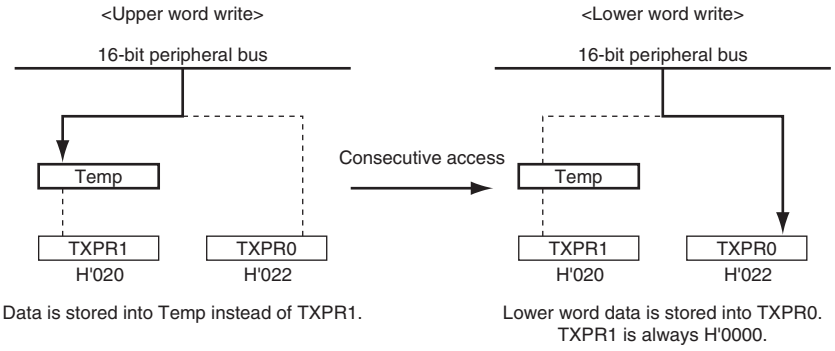
Description	Address	Name	Access Size (bits)
Transmit Pending 1	H'020	TXPR1	LW
Transmit Pending 0	H'022	TXPR0	—
	H'024		
	H'026		
	H'028		
Transmit Cancel 0	H'02A	TXCR0	
	H'02C		
	H'02E		
	H'030		
Transmit Acknowledge 0	H'032	TXACK0	Word
	H'034		
	H'036		
	H'038		
Abort Acknowledge 0	H'03A	ABACK0	Word
	H'03C		
	H'03E		
	H'040		
Data Frame Receive Pending 0	H'042	RXPR0	Word
	H'044		
	H'046		
	H'048		
Remote Frame Receive Pending 0	H'04A	RFPR0	Word
	H'04C		
	H'04E		
	H'050		

Description	Address	Name	Access Size (bits)
Mailbox Interrupt Mask Register 0	H'052	MBIMR0	Word
	H'054		
	H'056		
	H'058		
Unread Message Status Register 0	H'05A	UMSR0	Word
	H'05C		
	H'05E		

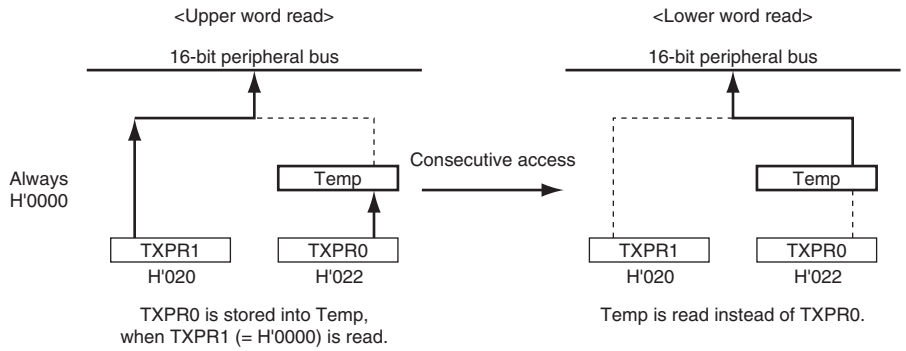
19.5.1 Transmit Pending Register (TXPR0, TXPR1)

The concatenation of TXPR0 and TXPR1 is a 32-bit register that contains any transmit pending flags for the CAN module. In the case of 16-bit bus interface, Long Word access is carried out as two consecutive word accesses.

<Longword Write Operation>



<Longword Read Operation>



The TXPR1 register cannot be modified and it is always fixed to '0'. The TXPR0 controls Mailbox-15 to Mailbox-1. The CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a '0' and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the CPU to determine which, if any, transmissions are pending or in progress. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is not configured to transmit is not allowed.

The RCAN-ET will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and RCAN-ET automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set), the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the RCAN-ET shall ensure that in the identifier priority scheme (MCR2 = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to section 19.6, Application Note, for details.

When the RCAN-ET changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR8) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signalled in the TXACK register, and if a message transmission abortion is successful it is signalled in the ABACK register. By checking these registers, the contents of the Message of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Any write operation is ignored.

Read value is always H'0000. Long word access is mandatory when reading or writing TXPR1/TXPR0. Writing any value to TXPR1 is allowed, however, write operation to TXPR1 has no effect. Writing to the bit 0 in TXPR0 has no effect.

- TXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	—

Note: * it is possible only to write a '1' for a Mailbox configured as transmitter.

Bit 15 to 1 — Indicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number.

Bit[15:1]:TXPR0	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of message transmission or message transmission abortion (automatically cleared)
1	Transmission request made for corresponding mailbox

Bit 0— Reserved: This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is '0'.

19.5.2 Transmit Cancel Register 0 (TXCR0)

TXCR0 is a 16-bit read/conditionally-write registers. The TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the CPU to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the CPU must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the CPU to clear a mailbox transmission that is not transmit-pending it has no effect. In this case the CPU will be not able at all to set the TXCR flag.

- TXCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	—

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.

Bit 15 to 1 — Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively.

Bit[15:1]:TXCR0	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing Condition] Completion of transmit message cancellation (automatically cleared)
1	Transmission cancellation request made for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

19.5.3 Transmit Acknowledge Register 0 (TXACK0)

The TXACK0 is a 16-bit read/conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been successfully made. When a transmission has succeeded the RCAN-ET sets the corresponding bit in the TXACK register. The CPU may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	—

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:TXACK0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) [Setting Condition] Completion of message transmission for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

19.5.4 Abort Acknowledge Register 0 (ABACK0)

The ABACK0 is a 16-bit read/conditionally-write registers. This register is used to signal to the CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded the RCAN-ET sets the corresponding bit in the ABACK register. The CPU may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the RCAN-ET to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	—

Note: * Only when writing a '1' to clear.

Bit 15 to 1 — Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively.

Bit[15:1]:ABACK0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) [Setting Condition] Completion of transmission cancellation for corresponding mailbox

Bit 0 — This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

19.5.5 Data Frame Receive Pending Register 0 (RXPR0)

The RXPR0 is a 16-bit read/conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames.

- RXPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Configurable receive mailbox locations corresponding to each mailbox position from 15 to 0 respectively.

Bit[15:0]: RXPR0	Description
0	[Clearing Condition] Writing '1' (Initial value)
1	Corresponding Mailbox received a CAN Data Frame [Setting Condition] Completion of Data Frame receive on corresponding mailbox

19.5.6 Remote Frame Receive Pending Register 0 (RFPR0)

The RFPR0 is a 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Request Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames.

- RFPR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: * Only when writing a '1' to clear.

Bit 15 to 0 — Remote Request pending flags for mailboxes 15 to 0 respectively.

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

19.5.7 Mailbox Interrupt Mask Register 0 (MBIMR0)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Request Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun/OverWrite Interrupt. If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or UMSR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, and it does not prevent the RCAN-ET from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[15:0]															
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

Bit 15 to 0 — Enable or disable interrupt requests from individual Mailbox-15 to Mailbox-0 respectively.

Bit[15:0]: MBIMR0	Description
0	Interrupt Request from IRR1/IRR2/IRR8/IRR9 enabled
1	Interrupt Request from IRR1/IRR2/IRR8/IRR9 disabled (initial value)

19.5.8 Unread Message Status Register 0 (UMSR0)

This register is a 16-bit read/conditionally write register and it records the mailboxes whose contents have not been accessed by the CPU prior to a new message being received. If the CPU has not cleared the corresponding bit in the RXPR or RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to '1'. This bit may be cleared by writing a '1' to the corresponding bit location in the UMSR. Writing a '0' has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit 15 to 0 — Indicate that an unread received message has been overwritten or overrun condition has occurred for Mailboxes 15 to 0.

Bit[15:0]: UMSR0	Description
0	[Clearing Condition] Writing '1' (initial value)
1	Unread received message is overwritten by a new message or overrun condition [Setting Condition] When a new message is received before RXPR or RFPR is cleared

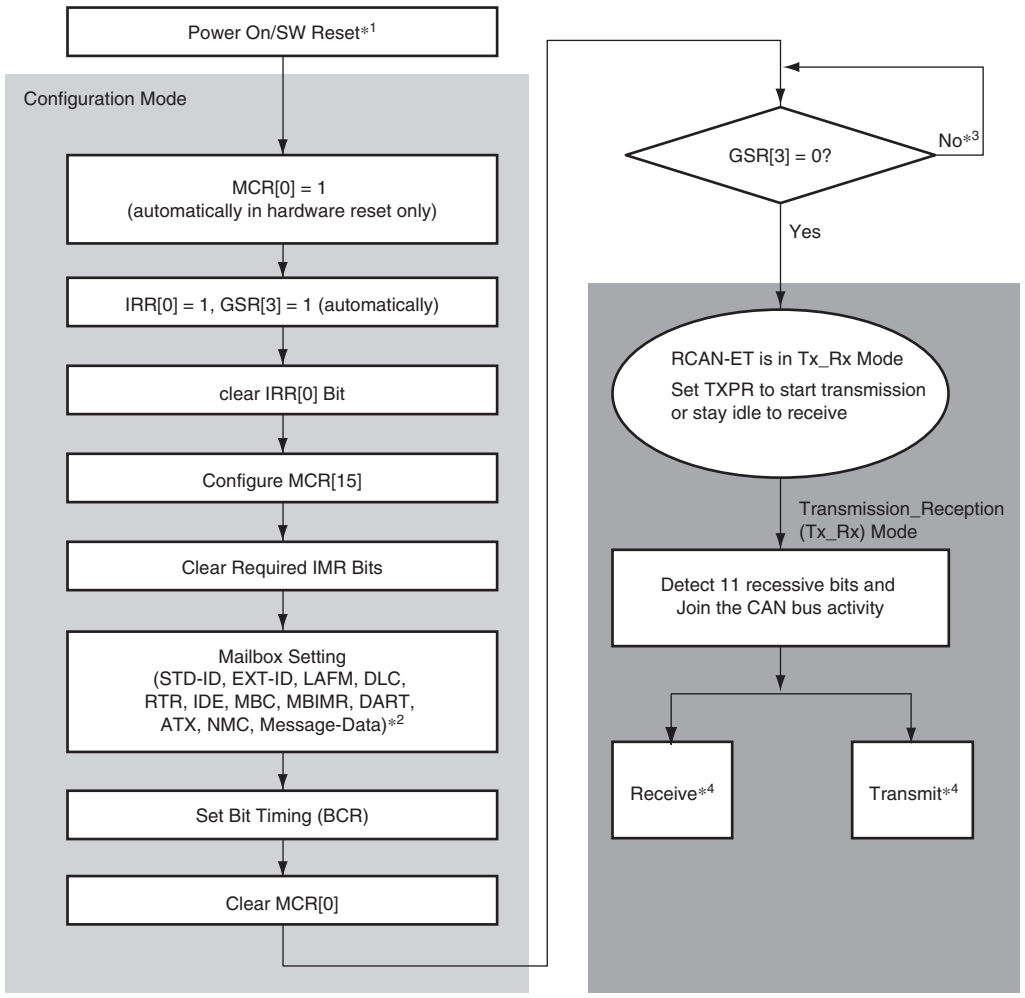
19.6 Application Note

19.6.1 Configuration of RCAN-ET

RCAN-ET is considered in configuration mode or after a H/W (Power On Reset)/ S/W (MCR[0]) reset or when in Halt mode. In both conditions RCAN-ET cannot join the CAN Bus activity and configuration changes have no impact on the traffic on the CAN Bus.

(1) After a reset request

The following sequence must be implemented to configure the RCAN-ET after (S/W or H/W) reset. After reset, all the registers are initialized, therefore, RCAN-ET needs to be configured before joining the CAN bus activity. Please read the notes carefully.



Notes: 1. SW reset could be performed at any time by setting MCR[0] = 1.
2. Mailboxes are comprised of RAMs, therefore, please initialize all the mailboxes enabled by MBC.
3. It takes approximately one bit time for GSR[3] to be cleared to 0.
4. If there is no TXPR set, RCAN-ET will receive the next incoming message.
If there is a TXPR(s) set, RCAN-ET will start transmission of the message and will be arbitrated by the CAN bus.
If it loses the arbitration, it will become a receiver.

Figure 19.6 Reset Sequence

(2) Halt mode

When RCAN-ET is in Halt mode, it cannot take part to the CAN bus activity. Consequently the user can modify all the requested registers without influencing existing traffic on the CAN Bus. It is important for this that the user waits for the RCAN-ET to be in halt mode before to modify the requested registers - note that the transition to Halt Mode is not always immediate (transition will occurs when the CAN Bus is idle or in intermission). After RCAN-ET transit to Halt Mode, GSR4 is set.

Once the configuration is completed the Halt request needs to be released. RCAN-ET will join CAN Bus activity after the detection of 11 recessive bits on the CAN Bus.

(3) Sleep mode

When RCAN-ET is in sleep mode the clock for the main blocks of the IP is stopped in order to reduce power consumption. Only the following user registers are clocked and can be accessed: MCR, GSR, IRR and IMR. Interrupt related to transmission (TXACK and ABACK) and reception (RXPR and RFPR) cannot be cleared when in sleep mode (as TXACK, ABACK, RXPR and RFPR are not accessible) and must to be cleared beforehand.

The following diagram shows the flow to follow to move RCAN-ET into sleep mode.

(4) CAN sleep mode

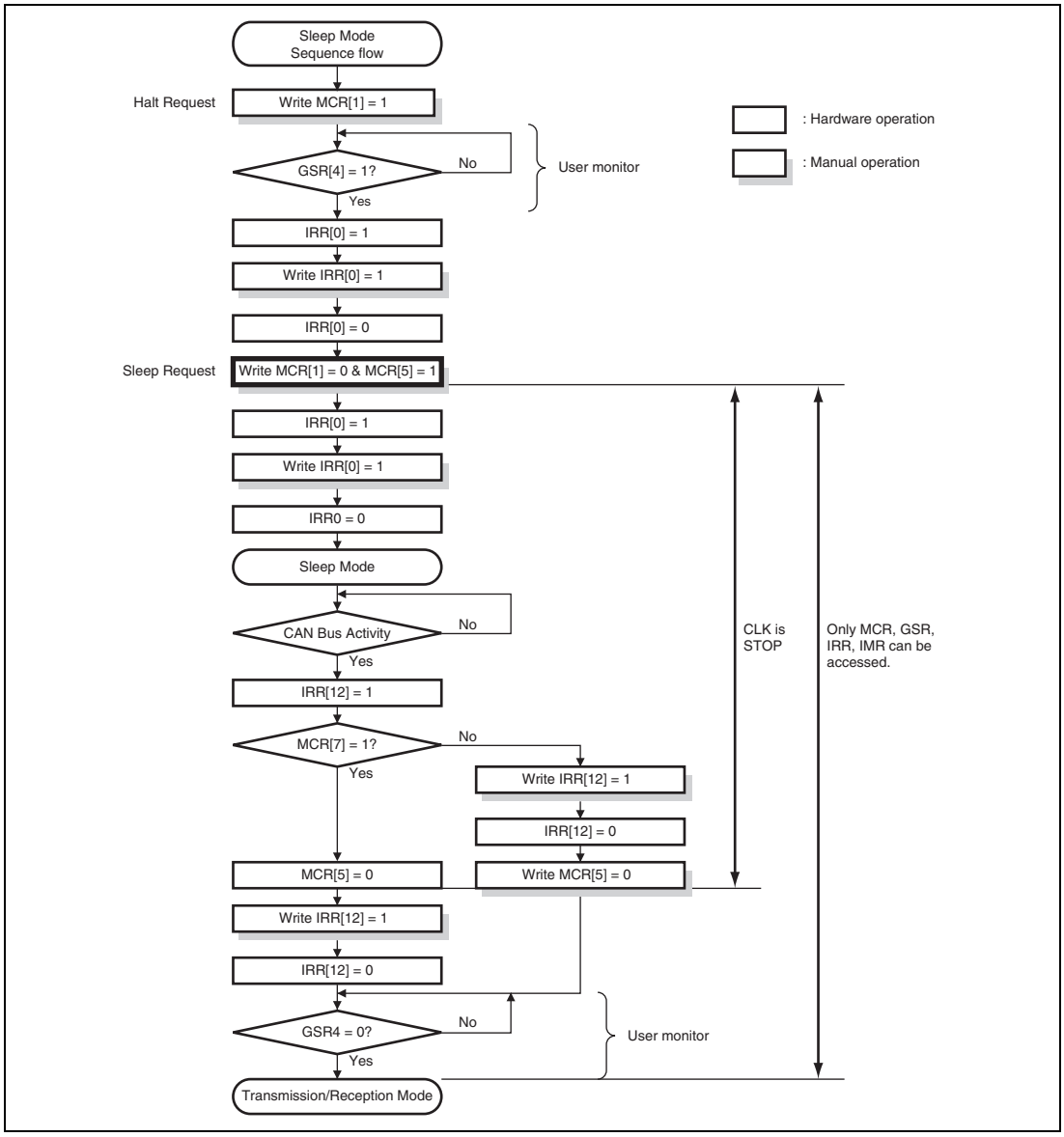
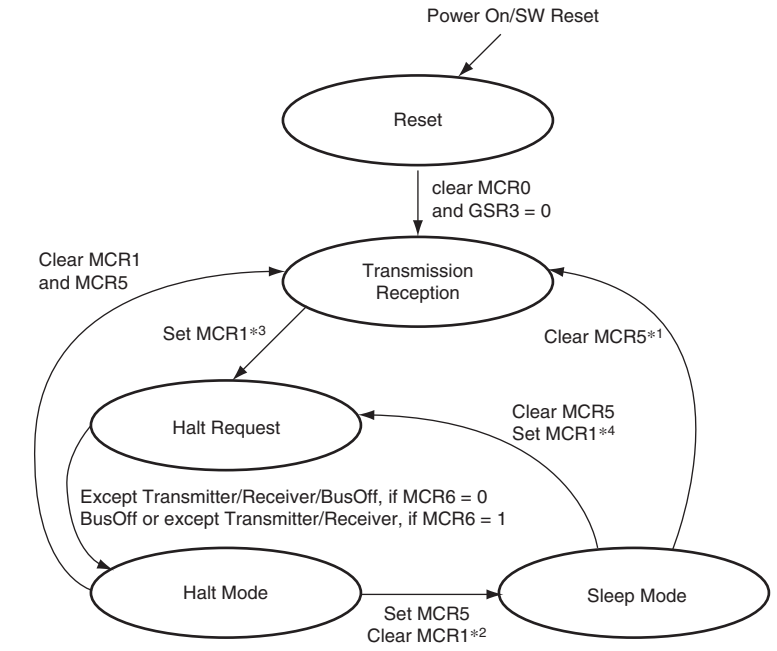


Figure 19.7 Halt Mode/Sleep Mode

Figure 19.8 - Halt Mode/Sleep Mode shows allowed state transition.

- Please don't set MCR5 (Sleep Mode) without entering Halt Mode.
- After setting MCR1, make sure that GSR4 is set and the RCAN-ET has entered Halt Mode before clearing MCR1.



Notes:
1. MCR5 can be cleared by automatically by detecting a dominant bit on the CAN Bus if MCR7 is set or by writing "0".
2. MCR1 is cleared in SW. Clearing MCR1 and setting MCR5 have to be carried out by the same instruction.
3. MCR1 must not be cleared in SW, before GSR4 is set. MCR1 can be set automatically in HW when RCAN-ET moves to Bus Off and MCR14 and MCR6 are both set.
4. When MCR5 is cleared and MCR1 is set at the same time, RCAN-ET moves to Halt Request. Right after that, it moves to Halt Mode with no reception/transmission.

Figure 19.8 Halt Mode/Sleep Mode

The following table shows conditions to access registers.

Table 19.8 Conditions to Access Registers

	RCAN-ET Registers								
Status Mode	MCR GSR	IRR IMR	BCR	MBIMR	Flag_register	mailbox (ctrl0, LAFM)		mailbox (data)	mailbox (ctrl1)	
Reset	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	
Transmission Reception	Yes	Yes	No*1	Yes	Yes	No*1	Yes*2	Yes*2	No*1	Yes*2
Halt Request	Yes	Yes	No*1	Yes	Yes	No*1	Yes*2	Yes*2	No*1	Yes*2
Halt	Yes	Yes	No*1	Yes	Yes	Yes		Yes	Yes	
Sleep	Yes	Yes	No	No	No	No		No	No	

Notes: 1. No hardware protection
2. When TXPR is not set.

19.6.2 Test Mode Settings

The RCAN-ET has various test modes. The register TST[2:0] (MCR[10:8]) is used to select the RCAN-ET test mode. The default (initialized) settings allow RCAN-ET to operate in Normal mode. The following table is examples for test modes.

Test Mode can be selected only while in configuration mode. The user must then exit the configuration mode (ensuring BCR0/BCR1 is set) in order to run the selected test mode.

Table 19.9 Test Mode Settings

Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	Normal mode (initial value)
0	0	1	Listen-only mode (receive-only mode)
0	1	0	Self test mode 1 (external)
0	1	1	Self test mode 2 (internal)
1	0	0	Write error counter
1	0	1	Error passive mode
1	1	0	Setting prohibited
1	1	1	Setting prohibited

- Normal Mode
  RCAN-ET operates in the normal mode.

- Listen-Only Mode:
  ISO-11898 requires this mode for baud rate detection. The Error Counters are cleared and disabled so that the TEC/REC does not increase the values, and the CTx Output is disabled so that RCAN-ET does not generate error frames or acknowledgment bits. IRR13 is set when a message error occurs.

- Self Test Mode 1
  RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRx/CTx pins must be connected to the CAN bus.

- Self Test Mode 2
  RCAN-ET generates its own Acknowledge bit, and can store its own messages into a reception mailbox (if required). The CRx/CTx pins do not need to be connected to the CAN bus or any external devices, as the internal CTx is looped back to the internal CRx. CTx pin outputs only recessive bits and CRx pin is disabled.

- Write Error Counter
  TEC/REC can be written in this mode. RCAN-ET can be forced to become an Error Passive mode by writing a value greater than 127 into the Error Counters. The value written into TEC is used to write into REC, so only the same value can be set to these registers. Similarly, RCAN-ET can be forced to become an Error Warning by writing a value greater than 95 into them.

- Error Passive mode
  RCAN-ET needs to be in Halt Mode when writing into TEC/REC (MCR1 must be "1" when writing to the Error Counter). Furthermore this test mode needs to be exited prior to leaving Halt mode.Error Passive Mode: RCAN-ET can be forced to enter Error Passive mode.

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

When message error occurs, IRR13 is set in all test modes.

19.6.3 Message Transmission Sequence

(1) Message Transmission Request

The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR8 is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR2 means that there is currently no transmission request made (No TXPR flags set).

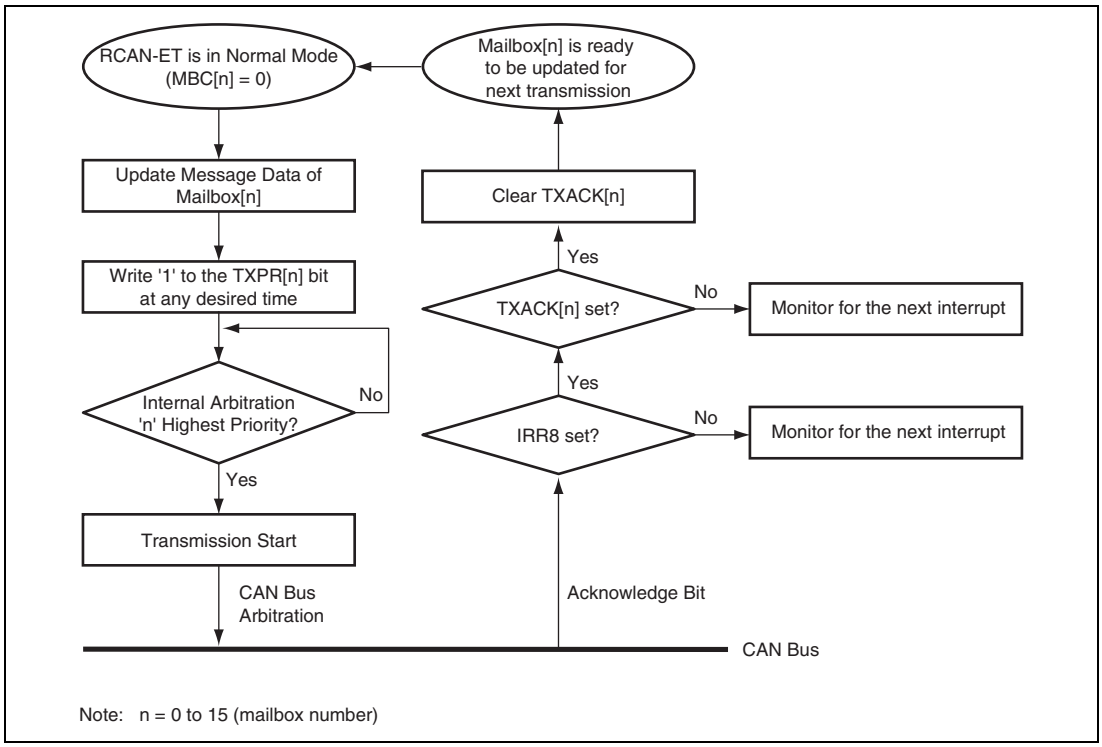


Figure 19.9 Transmission Request

(2) Internal Arbitration for Transmission

The following diagram explains how RCAN-ET manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

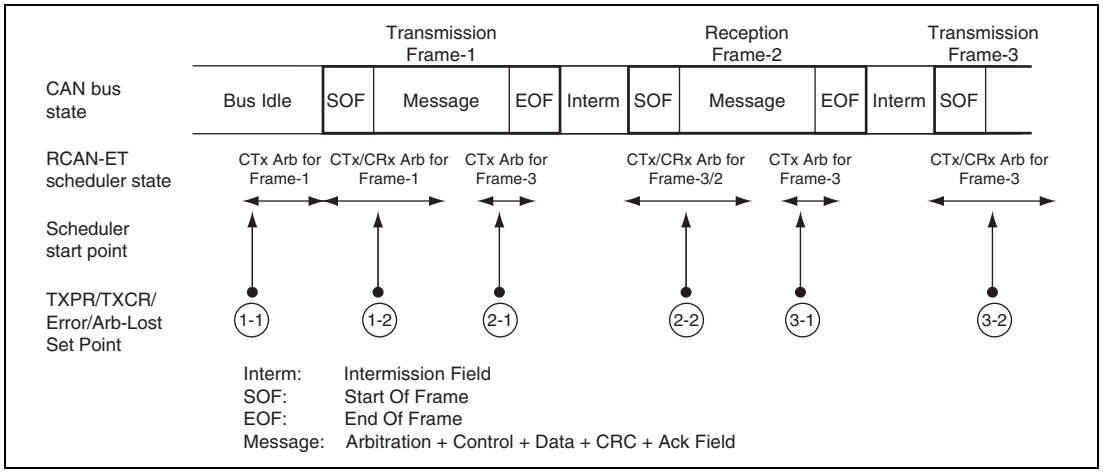


Figure 19.10 Internal Arbitration for transmission

The RCAN-ET has two state machines. One is for transmission, and the other is for reception.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

1-2: Operations for both transmission and reception starts at SOF. Since there is no reception frame, RCAN-ET becomes transmitter.

2-1: At crc delimiter, internal arbitration to search next message transmitted starts.

2-2: Operations for both transmission and reception starts at SOF. Because of a reception frame with higher priority, RCAN-ET becomes receiver. Therefore, Reception is carried out instead of transmitting Frame-3.

3-1: At crc delimiter, internal arbitration to search next message transmitted starts.

3-2: Operations for both transmission and reception starts at SOF. Since a transmission frame has higher priority than reception one, RCAN-ET becomes transmitter.

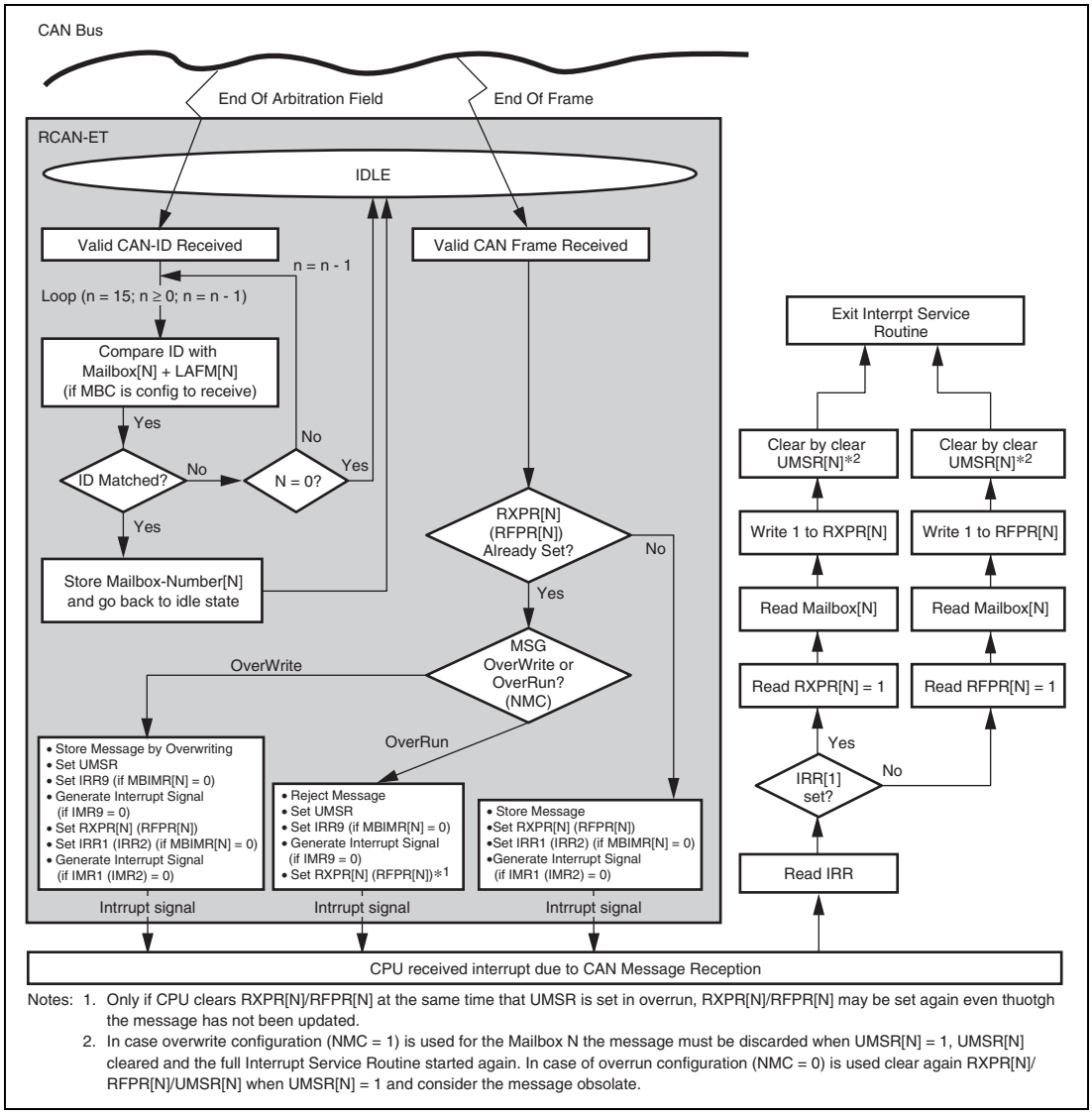
Internal arbitration for the next transmission is also performed at the beginning of each error delimiter in case of an error is detected on the CAN Bus. It is also performed at the beginning of error delimiters following overload frame.

As the arbitration for transmission is performed at CRC delimiter, in case a remote frame request is received into a Mailbox with ATX = 1 the answer can join the arbitration for transmission only at the following Bus Idle, CRC delimiter or Error Delimiter.

Depending on the status of the CAN bus, following the assertion of the TXCR, the corresponding Message abortion can be handled with a delay of maximum 1 CAN Frame.

19.6.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Notes: 1. Only if CPU clears RXPR[N]/RFPR[N] at the same time that UMSR is set in overrun, RXPR[N]/RFPR[N] may be set again even thuotgh the message has not been updated.
2. In case overwrite configuration (NMC = 1) is used for the Mailbox N the message must be discarded when UMSR[N] = 1, UMSR[N] cleared and the full Interrupt Service Routine started again. In case of overrun configuration (NMC = 0) is used clear again RXPR[N]/ RFPR[N]/UMSR[N] when UMSR[N] = 1 and consider the message obsolate.

Figure 19.11 Message receive sequence

When RCAN-ET recognises the end of the Arbitration field while receiving a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-15 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-15 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-14 (if configured as receive). Once RCAN-ET finds a matching identifier, it stores the number of Mailbox-[n] into an internal buffer, stops the search, and goes back to idle state, waiting for the EndOfFrame (EOF) to come. When the 6th bit of EOF is notified by the CAN Interface logic, the received message is written or abandoned, depending on the NMC bit. No modification of configuration during communication is allowed. Entering Halt Mode is one of ways to modify configuration. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

With regards to the reception of data and remote frames described in the above flow diagram the clearing of the UMSR flag after the reading of IRR is to detect situations where a message is overwritten by a new incoming message stored in the same mailbox while the interrupt service routine is running. If during the final check of UMSR a overwrite condition is detected the message needs to be discarded and read again.

Please note that in the case a received remote frame is overwritten by a data frame, both the remote frame request interrupt (IRR2) and data frame received interrupt (IRR1) and also the Receive Flags (RXPR and RFPR) are set. In an analogous way, the overwriting of a data frame by a remote frame, leads to setting both IRR2 and IRR1.

In the Overrun Mode (NMC = '0'), only the first Mailbox will cause the flags to be asserted. So, if a Data Frame is initially received, then RXPR and IRR1 are both asserted. If a Remote Frame is then received before the Data Frame has been read, then RFPR and IRR2 are NOT set. In this case UMSR of the corresponding Mailbox will still be set.

19.6.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

(1) Change configuration of transmit box

Two cases are possible.

- Change of ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART
  This change is possible only when MBC = B'000. Confirm that the corresponding TXPR is not set. The configuration (except MBC bit) can be changed at any time.
- Change from transmit to receive configuration (MBC)
  Confirm that the corresponding TXPR is not set. The configuration can be changed only in Halt or reset state. Please note that it might take longer for RCAN-ET to transit to halt state if it is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt state.
  In case RCAN-ET is in the Bus Off state the transition to halt state depends on the configuration of the bit 6 of MCR and also bit and 14 of MCR.

(2) Change configuration (ID, RTR, IDE, LAFM, Data, DLC, NMC, ATX, DART) of receive box or Change receive box to transmit box

The configuration can be changed only in Halt Mode.

RCAN-ET will not lose a message if the message is currently on the CAN bus and RCAN-ET is a receiver. RCAN-ET will be moving into Halt Mode after completing the current reception. Please note that it might take longer if RCAN-ET is receiving or transmitting a message (as the transition to the halt state is delayed until the end of the reception/transmission), and also RCAN-ET will not be able to receive/transmit messages during the Halt Mode.

In case RCAN-ET is in the Bus Off state the transition to halt mode depends on the configuration of the bit 6 and 14 of MCR.

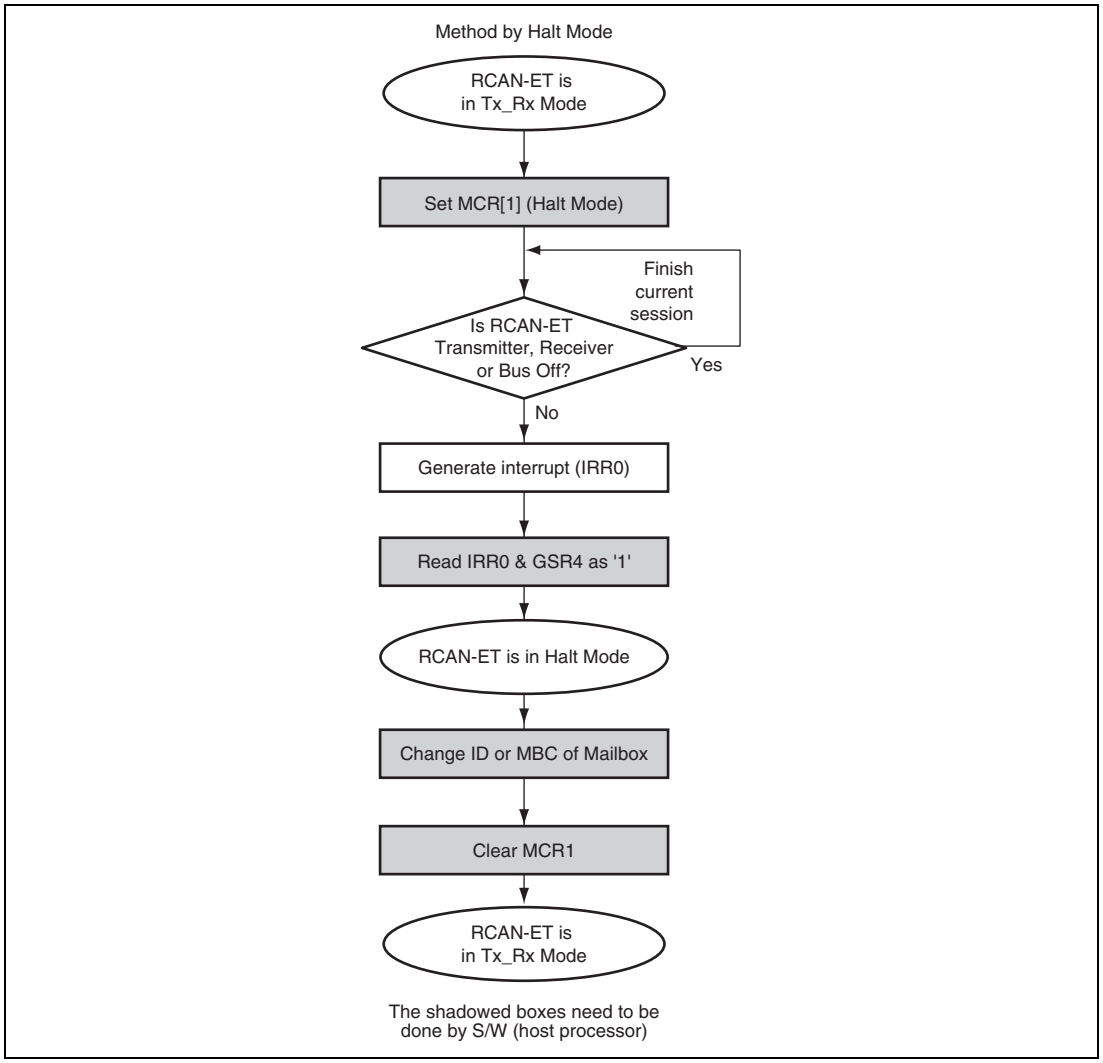


Figure 19.12 Change ID of Receive Box or Change Receive Box to Transmit Box

19.7 Interrupt Sources

Table 19.10 lists the RCAN-ET interrupt sources. With the exception of the reset processing interrupt (IRR0) by a power-on reset, these sources can be masked. Masking is implemented using the mailbox interrupt mask register 0 (MBIMR0) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 6, Interrupt Controller (INTC).

Table 19.10 RCAN-ET Interrupt Sources

Channel	Interrupt	Description	Interrupt Flag	DTC Activation
0	ERS_0	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
		Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
		Error warning (TEC ≥ 96)	IRR3	
		Error warning (REC ≥ 96)	IRR4	
	OVR_0	Message error detection	IRR13*[1]	
		Reset/halt/CAN sleep transition	IRR0	
		Overload frame transmission	IRR7	
		Unread message overwrite (overrun)	IRR9	
		Detection of CAN bus operation in CAN sleep mode	IRR12	
	SLE_0	Message transmission/transmission disabled (slot empty)	IRR8	
	RM1_0*[2]	Data frame reception/	IRR1*[3]	
	RM0_0*[2]	Remote frame reception	IRR2*[3]	Possible

Channel	Interrupt	Description	Interrupt Flag	DTC Activation
1	ERS_1	Error Passive Mode (TEC ≥ 128 or REC ≥ 128)	IRR5	Not possible
		Bus Off (TEC ≥ 256)/Bus Off recovery	IRR6	
		Error warning (TEC ≥ 96)	IRR3	
		Error warning (REC ≥ 96)	IRR4	
	OVR_1	Message error detection	IRR13*[1]	
		Reset/halt/CAN sleep transition	IRR0	
		Overload frame transmission	IRR7	
		Unread message overwrite (overrun)	IRR9	
		Detection of CAN bus operation in CAN sleep mode	IRR12	
	SLE_1	Message transmission/transmission disabled (slot empty)	IRR8	
	RM1_1*[2]	Data frame reception/	IRR1*[3]	
	RM0_1*[2]	Remote frame reception	IRR2*[3]	Possible

Notes: 1. Available only in Test Mode.

2. RM0 is an interrupt generated by the remote request pending flag for mailbox 0 (RFPR0[0]) or the data frame receive flag for mailbox 0 (RXPR0[0]). RM1 is an interrupt generated by the remote request pending flag for mailbox n (RFPR0[n]) or the data frame receive flag for mailbox n (RXPR0[n]) (n = 1 to 15).

3. IRR1 is a data frame received interrupt flag for mailboxes 0 to 15, and IRR2 is a remote frame request interrupt flag for mailboxes 0 to 15.

19.8 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. As the CRx and CTx pins use 3 V, an external level shifter is necessary. Figure 19.13 shows a sample connection diagram.

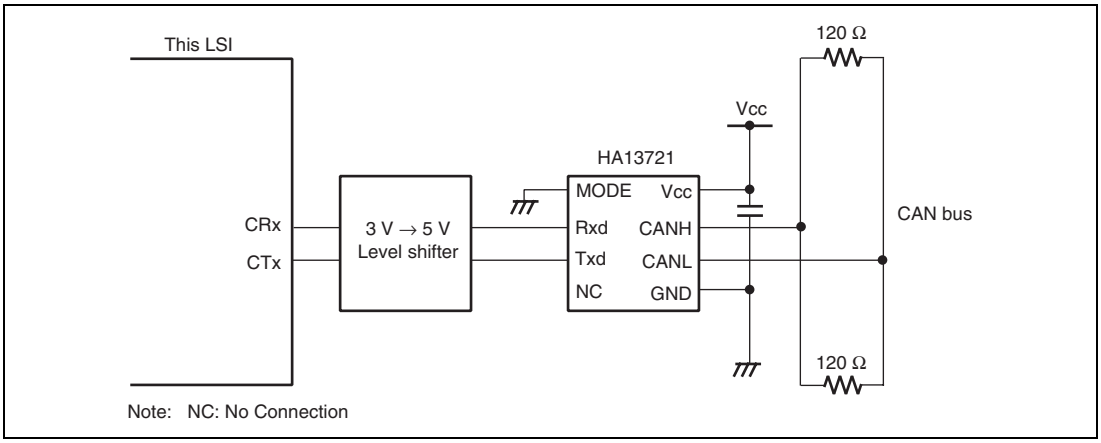


Figure 19.13 High-Speed Interface Using HA13721

19.9 Usage Notes

19.9.1 Module Standby Mode

The standby control register 2 (STBCR2) controls the supply of clocks to RCAN-ET. As an initial value, the clock to RCAN-ET is halted. Registers should be accessed after the module stop mode is released.

19.9.2 Reset

Two types of resets are supported for RCAN-ET.

- Hardware reset
  RCAN-ET is initialized by a power-on reset, deep standby mode, or software standby mode.
- Software reset
  The MCR0 bit in the master control register (MCR) initializes registers other than MCR and CAN communication functions.

As the IRR0 bit in the interrupt request register (IRR) is initialized and set to 1 at a reset, it should be cleared to 0 in the configuration mode shown in the reset sequence diagram.

The area except for the message control field 1 (CONTROL1) of Mailbox is consisted of RAM, and not initialized at a reset. After a power-on reset, all the Mailboxes should be initialized in the configuration mode shown in the reset sequence diagram.

19.9.3 CAN Sleep Mode

The supply of main clocks in the modules is stopped in CAN sleep mode. Therefore, registers other than MCR, GSR, IRR, and IMR should not be accessed in CAN sleep mode.

19.9.4 Register Access

When the CAN bus receive frame is being stored in the Mailbox with the CAN communication functions of RCAN-ET, accessing the Mailbox area generates 0 to 5 peripheral bus cycles as a wait.

19.9.5 Interrupts

As shown in table 19.2, the Mailbox 0 receive interrupt enables the DMAC activation. When an interrupt is specified as to be activated by the Mailbox 0 receive interrupt and cleared by the interrupt source at the DMA transfer, up to the message control field 1 (CONTROL1) of Mailbox 0 should be read using the block transfer mode.

Section 20 A/D Converter (ADC)

This LSI includes a 10-bit successive-approximation A/D converter allowing selection of up to eight analog input channels.

20.1 Features

- Resolution: 10 bits
- Input channels: 8
- Minimum conversion time: 3.9 μs per channel
- Operating modes: 3
  - Single mode: A/D conversion on one channel
  - Multi mode: A/D conversion on one to four channels or on one to eight channels
  - Scan mode: Continuous A/D conversion on one to four channels or on one to eight channels
- Data registers: 8

  Conversion results are held in a 16-bit data register for each channel
- Sample-and-hold function
- A/D conversion start methods: 4
  - Software
  - Conversion start trigger from multi-function timer pulse unit 2 (MTU2)
  - Conversion start trigger from the 8-bit timer (TMR)
  - External trigger signal
- Interrupt source

  An A/D conversion end interrupt (ADI) request can be generated on completion of A/D conversion.
- Module standby mode can be set

Figure 20.1 shows a block diagram of the A/D converter.

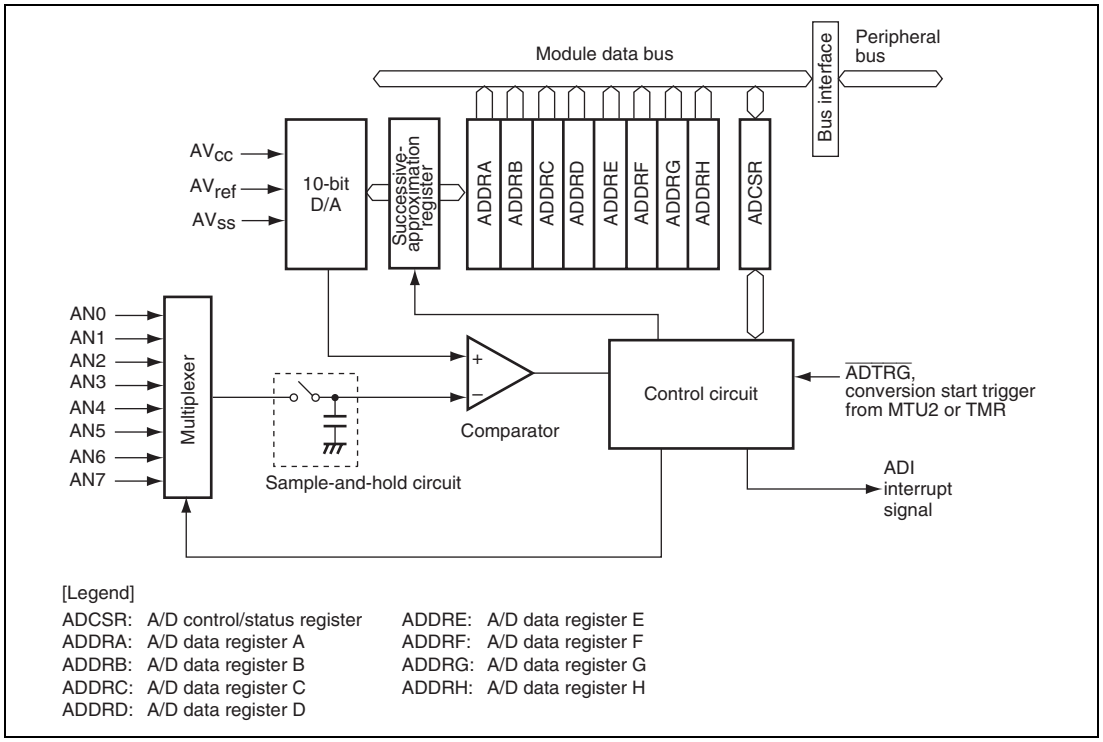


Figure 20.1 Block Diagram of A/D Converter

20.2 Input/Output Pins

Table 20.1 summarizes the A/D converter's input pins.

Table 20.1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVcc	Input	Analog power supply pin
Analog ground pin	AVss	Input	Analog ground pin and A/D conversion reference ground
Reference power supply pin	AVref	Input	A/D converter reference voltage pin
Analog input pin 0	AN0	Input	Analog input
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input to start A/D conversion

20.3 Register Configuration

The A/D converter has the following registers.

Table 20.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
A/D data register A	ADDRA	R	H'0000	H'FFFE5800	16
A/D data register B	ADDRB	R	H'0000	H'FFFE5802	16
A/D data register C	ADDRC	R	H'0000	H'FFFE5804	16
A/D data register D	ADDRD	R	H'0000	H'FFFE5806	16
A/D data register E	ADDRE	R	H'0000	H'FFFE5808	16
A/D data register F	ADDRF	R	H'0000	H'FFFE580A	16
A/D data register G	ADDRG	R	H'0000	H'FFFE580C	16
A/D data register H	ADDRH	R	H'0000	H'FFFE580E	16
A/D control/status register	ADCSR	R/W	H'0040	H'FFFE5820	16

20.3.1 A/D Data Registers A to H (ADDRA to ADDRH)

The sixteen A/D data registers, ADDRA to ADDRH, are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the ADDR corresponding to the selected channel. The 10 bits of the result are stored in the upper bits (bits 15 to 6) of ADDR. Bits 5 to 0 of ADDR are reserved bits that are always read as 0.

Access to ADDR in 8-bit units is prohibited. ADDR must always be accessed in 16-bit units.

ADDR is initialized to H'0000 by a power-on reset as well as in deep standby mode, software standby mode or module standby mode.

Table 20.3 indicates the pairings of analog input channels and ADDR.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 6		All 0	R	Bit data (10 bits)
5 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Table 20.3 Analog Input Channels and ADDR

Analog Input Channel	A/D Data Register where Conversion Result is Stored
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

20.3.2 A/D Control/Status Register (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode, controls the A/D converter, and enables or disables starting of A/D conversion by external trigger input.

ADCSR is initialized to H'0040 by a power-on reset as well as in deep standby mode, software standby mode or module standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	—	TRGS[3:0]				CKS[1:0]		MDS[2:0]			CH[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R/(W)*	R/W	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

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

Bit	Bit Name	Initial Value	R/W	Description
15	ADF	0	R/(W)*[1]	A/D End Flag Status flag indicating the end of A/D conversion. [Clearing conditions] • Cleared by reading ADF while ADF = 1, then writing 0 to ADF • Cleared when DMAC is activated by ADI interrupt and ADDR is read [Setting conditions] • A/D conversion ends in single mode • A/D conversion ends for the selected channels in multi mode • A/D conversion ends for the selected channels in scan mode
14	ADIE	0	R/W	A/D Interrupt Enable Enables or disables the interrupt (ADI) requested at the end of A/D conversion. Set the ADIE bit while A/D conversion is not being made. 0: A/D end interrupt request (ADI) is disabled 1: A/D end interrupt request (ADI) is enabled

Bit	Bit Name	Initial Value	R/W	Description
13	ADST	0	R/W	A/D Start Starts or stops A/D conversion. This bit remains set to 1 during A/D conversion. 0: A/D conversion is stopped 1: Single mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion ends on the selected channel. Multi mode: A/D conversion starts. This bit is automatically cleared to 0 when A/D conversion is completed cycling through the selected channels. Scan mode: A/D conversion starts. A/D conversion is continuously performed until this bit is cleared to 0 by software, by a power-on reset as well as by a transition to deep standby mode, software standby mode or module standby mode.
12	—	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
11 to 8	TRGS[3:0]	0000	R/W	Timer Trigger Select These bits enable or disable starting of A/D conversion by a trigger signal. 0000: Start of A/D conversion by external trigger input is disabled 0001: A/D conversion is started by conversion trigger TRGAN from MTU2 0010: A/D conversion is started by conversion trigger TRG0N from MTU2 0011: A/D conversion is started by conversion trigger TRG4AN from MTU2 0100: A/D conversion is started by conversion trigger TRG4BN from MTU2 0101: Setting prohibited 0110: Setting prohibited 0111: Setting prohibited 1000: Setting prohibited 1001: A/D conversion is started by $\overline{\text{ADTRG}}$ 1010: A/D conversion is started by conversion trigger from the TMR 1011 to 1111: Setting prohibited
7, 6	CKS[1:0]	01	R/W	Clock Select These bits select the A/D conversion time.*[2] Set the A/D conversion time while A/D conversion is halted (ADST = 0). 00: Conversion time = 138 states (maximum) 01: Conversion time = 274 states (maximum) 10: Conversion time = 546 states (maximum) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	MDS[2:0]	000	R/W	Multi-scan Mode These bits select the operating mode for A/D conversion. 0xx: Single mode 100: Multi mode: A/D conversion on 1 to 4 channels 101: Multi mode: A/D conversion on 1 to 8 channels 110: Scan mode: A/D conversion on 1 to 4 channels 111: Scan mode: A/D conversion on 1 to 8 channels
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bits in ADCSR select the analog input channels.

MDS = 0xx	MDS = 100 or MDS = 110	MDS = 101 or MDS = 111
000: AN0	000: AN0	000: AN0
001: AN1	001: AN0, AN1	001: AN0, AN1
010: AN2	010: AN0 to AN2	010: AN0 to AN2
011: AN3	011: AN0 to AN3	011: AN0 to AN3
100: AN4	100: AN4	100: AN0 to AN4
101: AN5	101: AN4, AN5	101: AN0 to AN5
110: AN6	110: AN4 to AN6	110: AN0 to AN6
111: AN7	111: AN4 to AN7	111: AN0 to AN7

[Legend]

x: Don't care

Notes: 1. Only 0 can be written to clear the flag after 1 is read.
Please note that ADF flag becomes "0" in the following cases, too.
(1) Reading the state of ADF = 1 with CPU.
(2) Clearing ADF flag by having read ADDR with DMAC
(3) Set of ADF flag according to A/D conversion end
(4) Writing 0 in the ADF flag with CPU
2. To satisfy the absolute accuracy of the A/D converter characteristics, set a value greater than the minimum conversion time.

20.4 Operation

The A/D converter uses the successive-approximation method, and the resolution is 10 bits. It has three operating modes: single mode, multi mode, and scan mode. Switching the operating mode or analog input channels must be done while the ADST bit in ADCSR is 0 to prevent incorrect operation. The ADST bit can be set at the same time as the operating mode or analog input channels are changed.

20.4.1 Single Mode

Single mode should be selected when only A/D conversion on one channel is required.

In single mode, A/D conversion is performed once for the specified one analog input channel, as follows:

1. A/D conversion for the selected channel starts when the ADST bit in ADCSR is set to 1 by software, MTU2, TMR, or external trigger input.
2. When A/D conversion is completed, the A/D conversion result is transferred to the A/D data register corresponding to the channel.
3. After A/D conversion has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel selection is switched.

Typical operations when a single channel (AN1) is selected in single mode are described next. Figure 20.2 shows a timing diagram for this example (the bits which are set in this example belong to ADCSR).

1. Single mode is selected, input channel AN1 is selected (CH2 = 0, CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the A/D conversion result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADF = 1, and then writes 0 to the ADF flag.
6. The routine reads and processes the A/D conversion result (ADDRB).
7. Execution of the A/D interrupts handling routine ends. Then, when the ADST bit is set to 1, A/D conversion starts and steps 2. to 7. are executed.

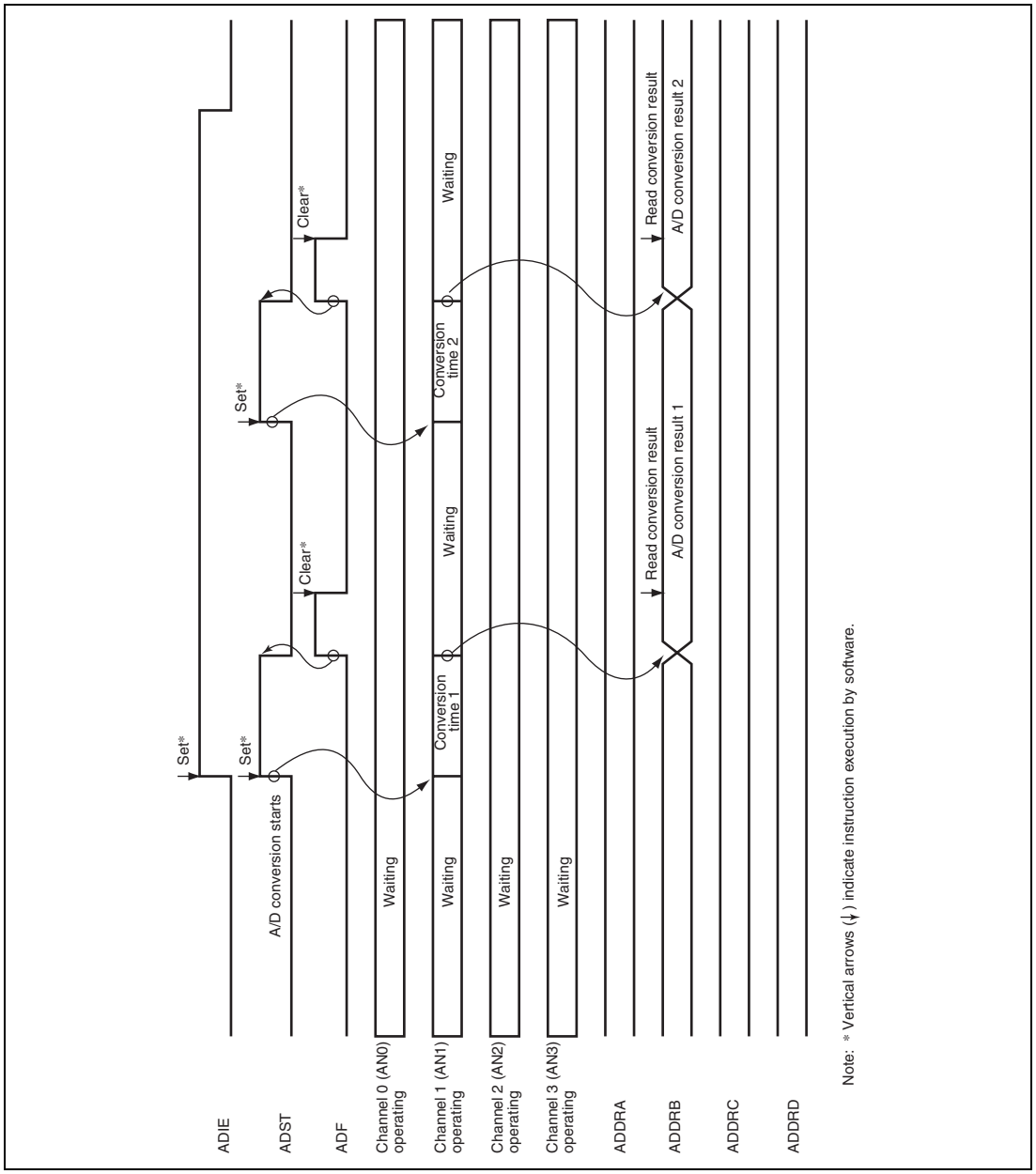


Figure 20.2 Example of A/D Converter Operation (Single Mode, One Channel (AN1) Selected)

20.4.2 Multi Mode

Multi mode should be selected when performing A/D conversion once on one or more channels.

In multi mode, A/D conversion is performed once for a maximum of eight specified analog input channels, as follows:

1. A/D conversion starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, MTU2, TMR, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit that remains 1 during A/D conversion is automatically cleared to 0 when A/D conversion is completed, and the A/D converter becomes idle. If the ADST bit is cleared to 0 during A/D conversion, A/D conversion is halted and the A/D converter becomes idle. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

A/D conversion is to be performed once on all the specified channels. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in multi mode are described next. Figure 20.3 shows a timing diagram for this example.

1. Multi mode is selected (MDS2 = 1, MDS1 = 0), analog input channels AN0 to AN2 are selected (CH2 = 0, CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and the ADST bit cleared to 0.
   If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

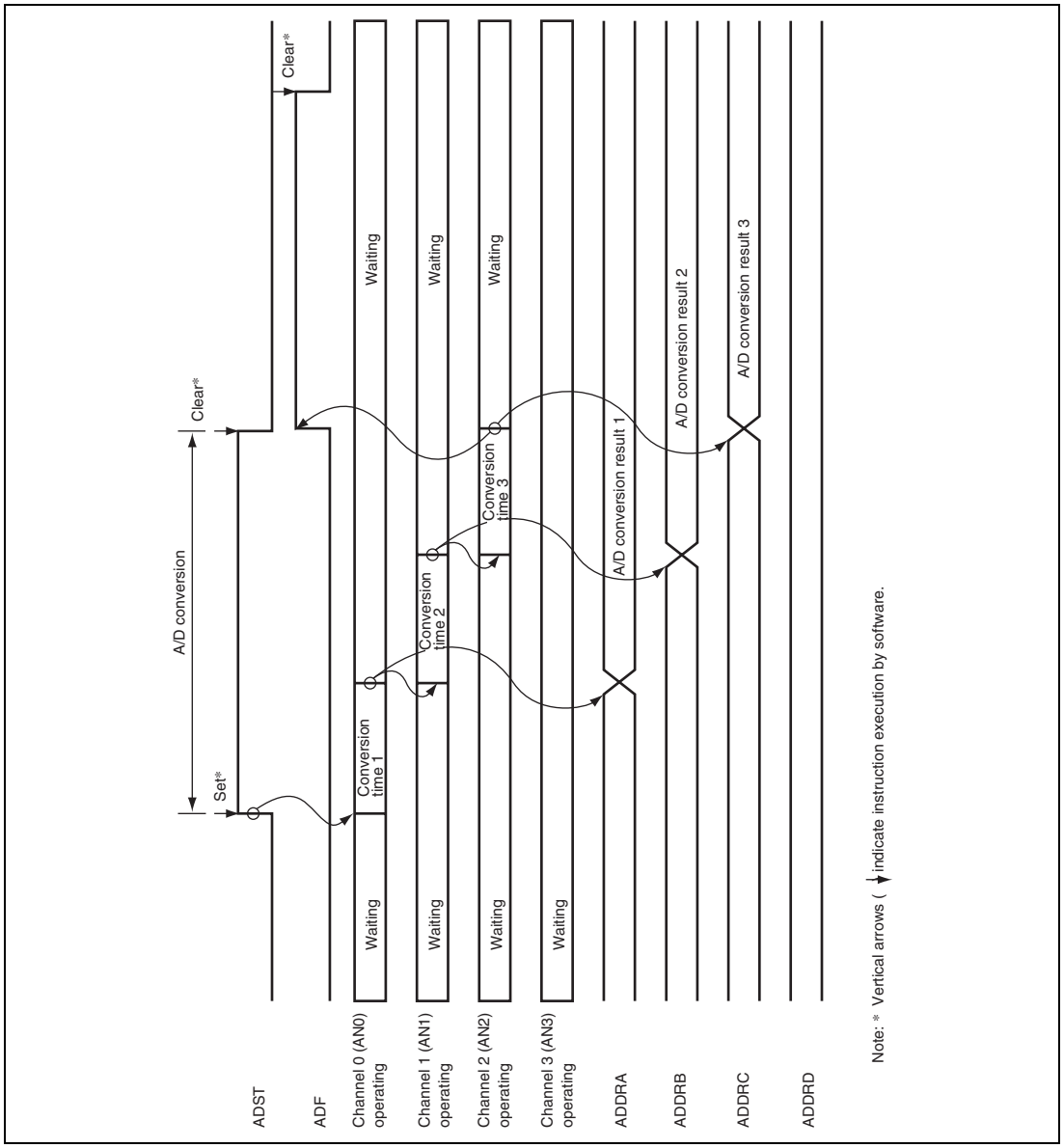


Figure 20.3 Example of A/D Converter Operation (Multi Mode, Three Channels (AN0 to AN2) Selected)

20.4.3 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels at all times. In scan mode, A/D conversion is performed sequentially for a maximum of eight specified analog input channels, as follows:

1. A/D conversion for the selected channels starts from the analog input channel with the lowest number (e.g. AN0, AN1, …, AN3) when the ADST bit in ADCSR is set to 1 by software, MTU2, TMR, or external trigger input.
2. When A/D conversion is completed on each channel, the A/D conversion result is sequentially transferred to the A/D data register corresponding to that channel.
3. After A/D conversion on all selected channels has completed, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The A/D converter starts A/D conversion again from the channel with the lowest number.
4. The ADST bit is not cleared automatically, so steps 2. and 3. are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion halts and the A/D converter becomes idle.

   The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

When the operating mode or analog input channel selection must be changed during A/D conversion, to prevent incorrect operation, first clear the ADST bit to 0 to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described as follows. Figure 20.4 shows a timing diagram for this example.

1. Scan mode is selected (MDS2 = 1, MDS1 = 1), analog input channels AN0 to AN2 are selected (CH2 = 0, CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. A/D conversion of the first channel (AN0) starts. When A/D conversion is completed, the A/D conversion result is transferred into ADDRA.
3. Next, the second channel (AN1) is selected automatically and A/D conversion starts.
4. Conversion proceeds in the same way through the third channel (AN2).
5. When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested.

6. The ADST bit is not cleared automatically, so steps 2. to 4. are repeated as long as the ADST bit remains set to 1. When steps 2. to 4. are repeated, the ADF flag is kept to 1. When the ADST bit is cleared to 0, A/D conversion stops. The ADF bit is cleared by reading ADF while ADF = 1, then writing 0 to the ADF bit.

If both the ADF flag and ADIE bit are set to 1 while steps 2. to 4. are repeated, an ADI interrupt is requested at all times. To generate an interrupt on completing conversion of the third channel, clear the ADF bit to 0 after an interrupt is requested.

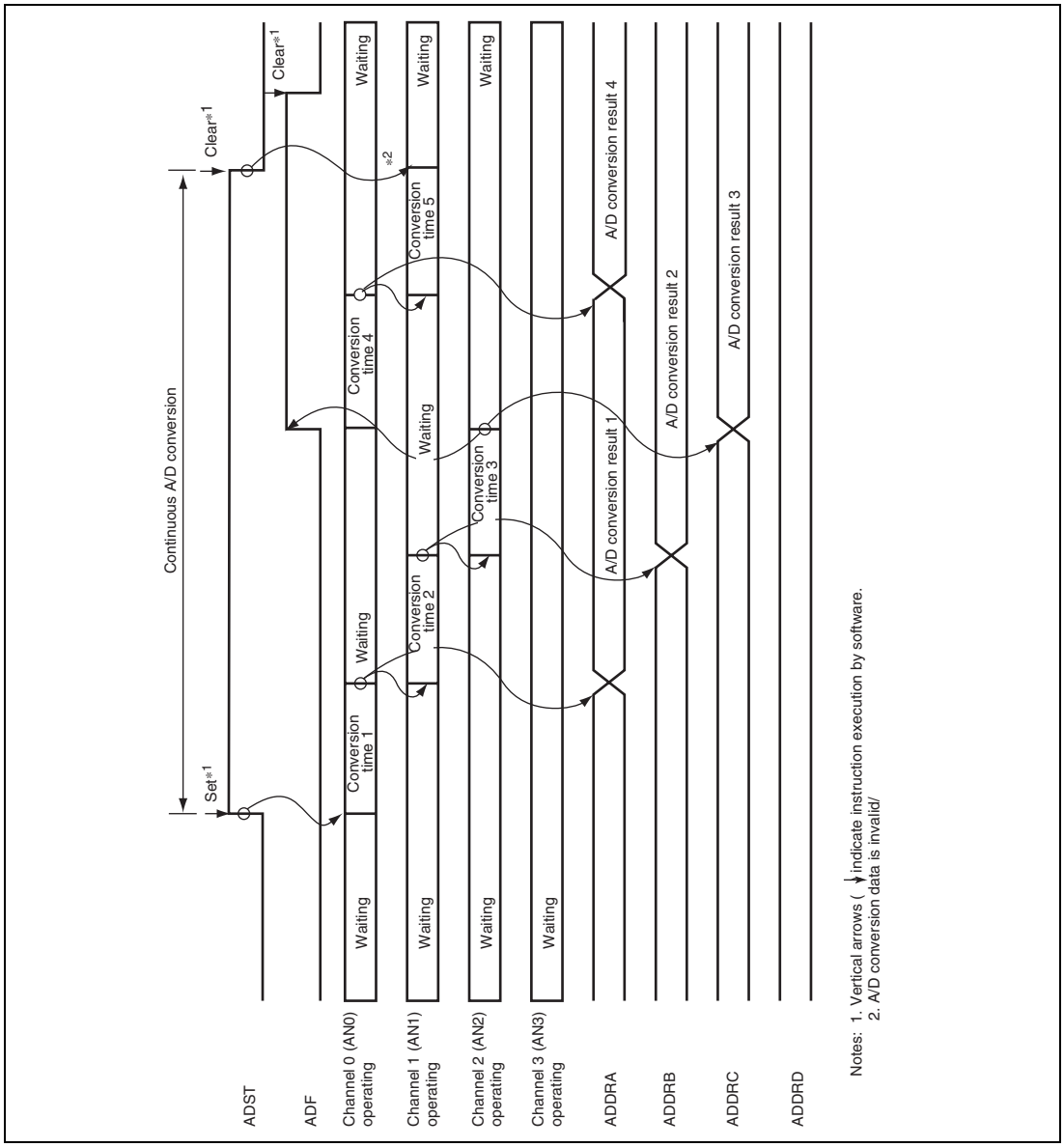


Figure 20.4 Example of A/D Converter Operation (Scan Mode, Three Channels (AN0 to AN2) Selected)

20.4.4 A/D Converter Activation by External Trigger, MTU2, or TMR

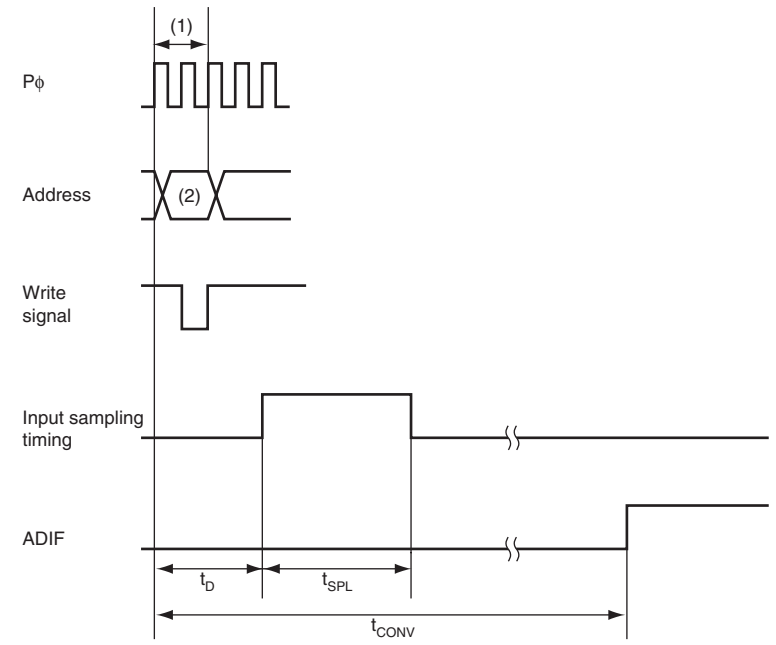
The A/D converter can be independently activated by an A/D conversion request from the external trigger, MTU2, or TMR. To activate the A/D converter by the external trigger, MTU2, or TMR, set the A/D trigger enable bits (TRGS[3:0]). After this bit setting has been made, the ADST bit is automatically set to 1 and A/D conversion is started when an A/D conversion request from the external trigger, MTU2, or TMR occurs. The channel combination is determined by the CH[2:0] bits in ADCSR. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written to the ADST bit by software.

20.4.5 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at the A/D conversion start delay time (t_D) after the ADST bit in ADCSR is set to 1, then starts conversion. Figure 20.5 shows the A/D conversion timing. Table 20.4 indicates the A/D conversion time.

As indicated in figure 20.5, the A/D conversion time (t_{CONV}) includes t_D and the input sampling time(t_{SPL}). The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 20.4.

In multi mode and scan mode, the values given in table 20.4 apply to the first conversion. In the second and subsequent conversions, time is the values given in table 20.5.



[Legend]
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay time
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 20.5 A/D Conversion Timing

Table 20.4 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 1		
		CKS0 = 0			CKS0 = 1			CKS0 = 0		
Item	**Symbol**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**	**Min.**	**Typ.**	**Max.**
A/D conversion start delay time	t_D	11	—	14	19	—	26	35	—	50
Input sampling time	t_{SPL}	—	33	—	—	65	—	—	129	—
A/D conversion time	t_{CONV}	135	—	138	267	—	274	531	—	546

Note: Values in the table are the numbers of states.

Table 20.5 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (States)
0	0	128 (constant)
	1	256 (constant)
1	0	512 (constant)

Note: Values in the table are the numbers of states.

20.4.6 External Trigger Input Timing

A/D conversion can also be externally triggered. When the TRGS[3:0] bits in ADCSR are set to B'1001, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, regardless of the operating mode, are the same as when the ADST bit has been set to 1 by software. Figure 20.6 shows the timing.

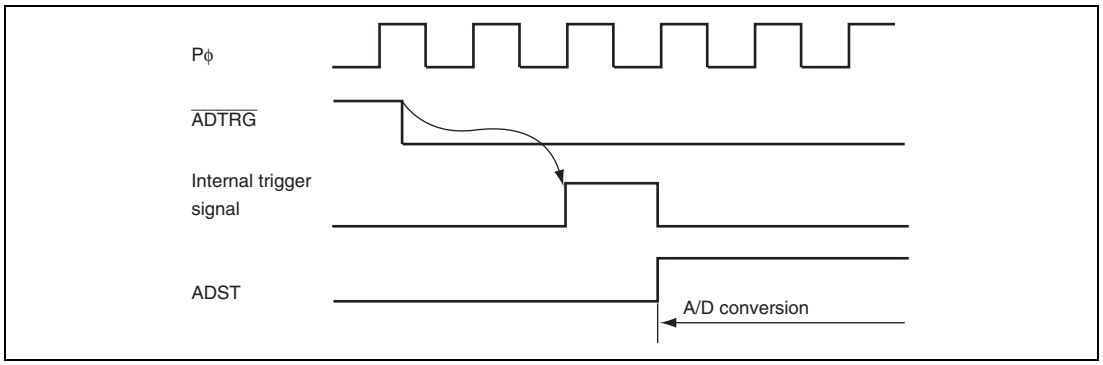


Figure 20.6 External Trigger Input Timing

20.5 Interrupt Sources and DMAC Transfer Request

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. An ADI interrupt request is generated if the ADIE bit is set to 1 when the ADF bit in ADCSR is set to 1 on completion of A/D conversion. Note that the direct memory access controller (DMAC) can be activated by an ADI interrupt depending on the interrupt controller (INTC) setting. 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. Having the converted data read by the DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

In single mode, set the DMAC so that DMA transfer initiated by an ADI interrupt is performed only once. In the case of A/D conversion on multiple channels in scan mode or multi mode, setting the DMA transfer count to one causes DMA transfer to finish after transferring only one channel of data. To make the DMAC transfer all conversion data, set the ADDR where A/D conversion data is stored as the transfer source address, the number of converted channels × 2 as the transfer byte count, and continuous operand transfer or non-stop transfer as the DMA transfer condition.

When the DMAC is activated by ADI, the ADF bit in ADCSR is automatically cleared to 0 when data is transferred by the DMAC.

Table 20.6 Relationship between Interrupt Sources and DMAC Transfer Request

Name	Interrupt Source	Interrupt Flag	DMAC Activation
ADI	A/D conversion end	ADF in ADCSR	Possible

20.6 Definitions of A/D Conversion Accuracy

The A/D converter compares an analog value input from an analog input channel with its analog reference value and converts it to 10-bit digital data. The absolute accuracy of this A/D conversion is the deviation between the input analog value and the output digital value. It includes the following errors:

- Offset error
- Full-scale error
- Quantization error
- Nonlinearity error

These four error quantities are explained below with reference to figure 20.7. In the figure, the 10 bits of the A/D converter have been simplified to 3 bits. Offset error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from the minimum (zero voltage) B'0000000000 (000 in the figure) to B'000000001 (001 in the figure) (figure 20.7, item (1)). Full-scale error is the deviation between actual and ideal A/D conversion characteristics when the digital output value changes from B'1111111110 (110 in the figure) to the maximum B'1111111111 (111 in the figure) (figure 20.7, item (2)). Quantization error is the intrinsic error of the A/D converter and is expressed as 1/2 LSB (figure 20.7, item (3)). Nonlinearity error is the deviation between actual and ideal A/D conversion characteristics between zero voltage and full-scale voltage (figure 20.7, item (4)). Note that it does not include offset, full-scale, or quantization error.

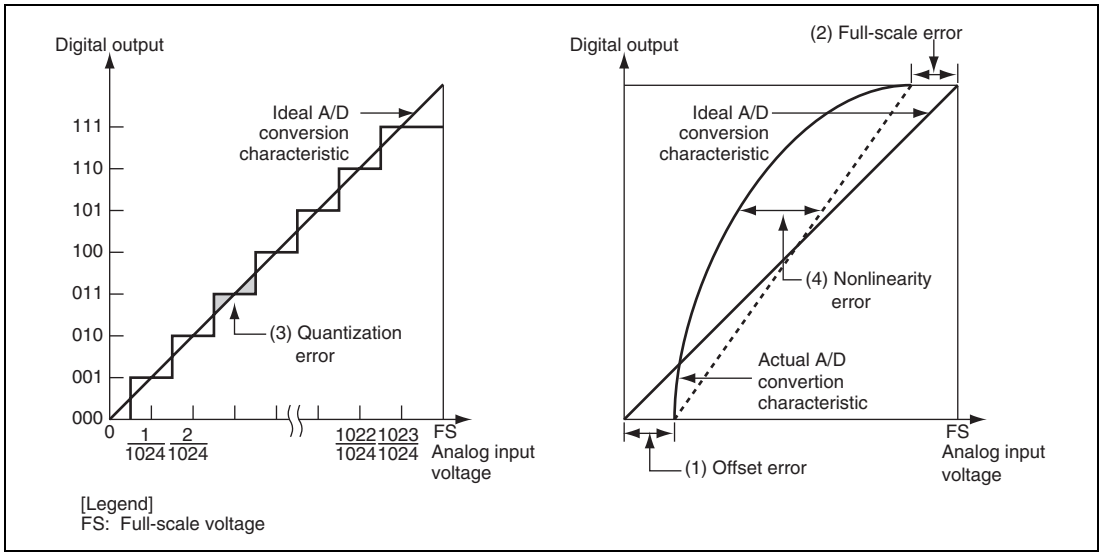


Figure 20.7 Definitions of A/D Conversion Accuracy

20.7 Usage Notes

When using the A/D converter, note the following points.

20.7.1 Module Standby Mode Setting

Operation of the A/D converter can be disabled or enabled using the standby control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module standby mode. For details, see section 25, Power-Down Modes.

20.7.2 Setting Analog Input Voltage

Permanent damage to the LSI may result if the following voltage ranges are exceeded.

1. Analog input range
   During A/D conversion, voltages on the analog input pins ANn should not go beyond the following range: AVss ≤ ANn ≤ AVcc (n = 0 to 7).
2. AVcc and AVss input voltages
   Input voltages AVcc and AVss should be PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = PVss. 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 (PVcc) and AVss to the ground (PVss).
3. Setting range of AVref input voltage
   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

20.7.3 Notes on Board Design

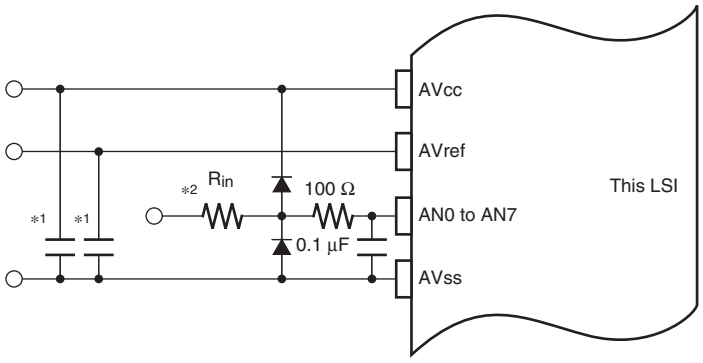
In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog reference voltage (AVref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (PVss) on the board.

20.7.4 Processing of Analog Input Pins

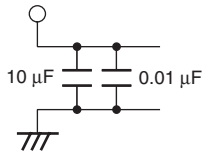
To prevent damage from voltage surges at the analog input pins (AN0 to AN7), connect an input protection circuit like the one shown in figure 20.8. The circuit shown also includes a CR filter to suppress noise. This circuit is shown as an example; the circuit constants should be selected according to actual application conditions.

Figure 20.9 shows an equivalent circuit diagram of the analog input ports and table 20.7 lists the analog input pin specifications.



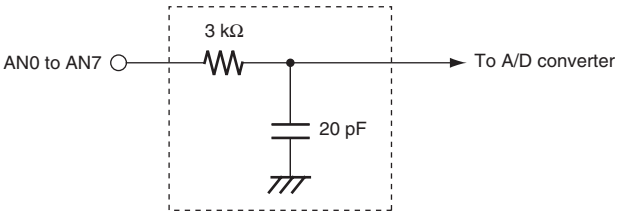
Notes: Values are reference values.

1.



2. R_{in}: Input impedance

Figure 20.8 Example of Analog Input Protection Circuit



Note: Values are reference values.

Figure 20.9 Analog Input Pin Equivalent Circuit

Table 20.7 Analog Input Pin Ratings

Item	Min.	Max.	Unit
Analog input capacitance	—	20	pF
Allowable signal-source impedance	—	5	kΩ

20.7.5 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 5 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 5 kΩ, charging may be insufficient and it may not be possible to guarantee A/D conversion precision. However, for A/D conversion in single mode with a large capacitance provided externally for A/D conversion in single mode, the input load will essentially comprise only the internal input resistance of 3 kΩ, and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/μs or greater) (see figure 20.10). When converting a high-speed analog signal, a low-impedance buffer should be inserted.

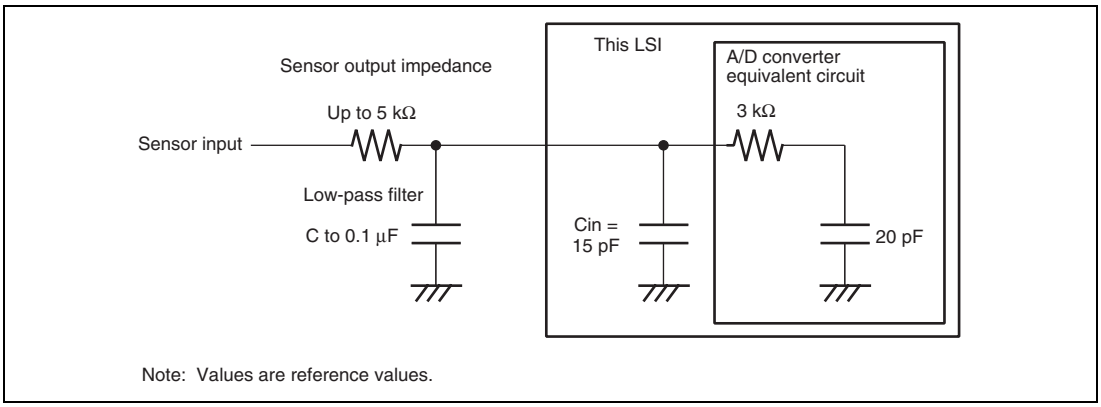


Figure 20.10 Example of Analog Input Circuit

20.7.6 Influences on Absolute Precision

Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to connect AVss, etc. to an electrically stable GND.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board (i.e., acting as antennas).

20.7.7 Note on Usage in Scan Mode and Multi Mode

Starting conversion immediately after having stopped scan mode or multi mode operation may lead to erroneous results of conversion.

To perform continuous conversion in such cases, set ADST to 0, wait for at least the A/D conversion time for a single channel to elapse, and then start conversion (ADST = 1). (The A/D conversion time for a single channel will vary according to the settings of the ADC registers.)

Section 21 D/A Converter (DAC)

21.1 Features

- Resolution: 8 bits
- Input channels: 2
- Minimum conversion time: Max.10 μs (with 20 pF load)
- Output voltage: 0 V to AVref
- D/A output hold function in software standby modes
- Module standby mode can be set

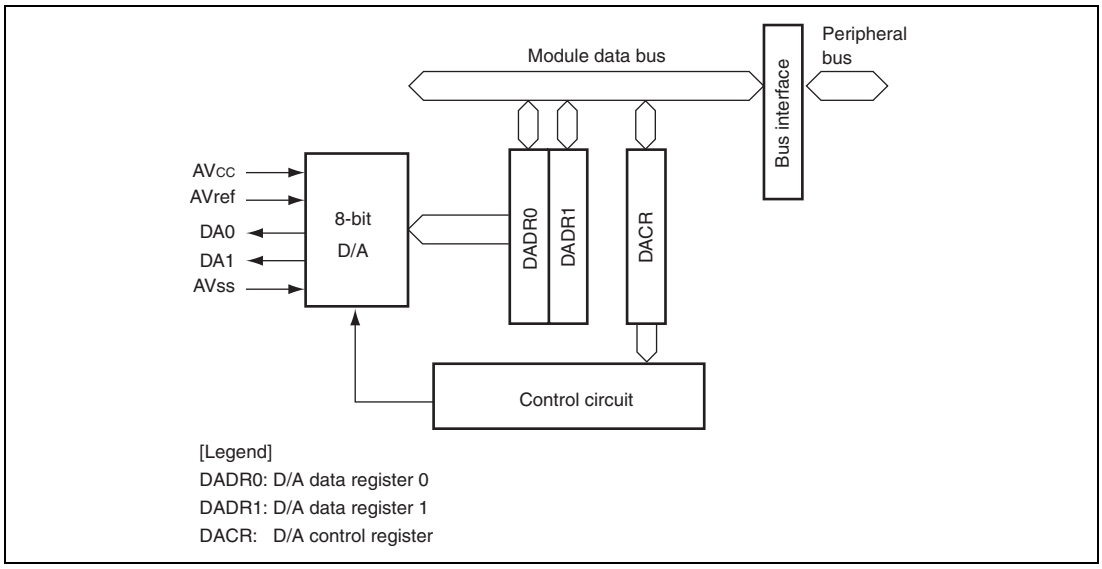


Figure 21.1 Block Diagram of D/A Converter

21.2 Input/Output Pins

Table 21.1 shows the pin configuration of the D/A converter.

Table 21.1 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
Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output

21.3 Register Descriptions

The D/A converter has the following registers.

Table 21.2 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

21.3.1 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 in deep standby mode or 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

21.3.2 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 in deep standby mode or 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.

Bit	Bit Name	Initial Value	R/W	Description
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 21.3. 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 21.3 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	

21.4 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 21.2 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:

$$\frac{\text{Contents of DADR}}{256} \quad \text{AVref}$$

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.

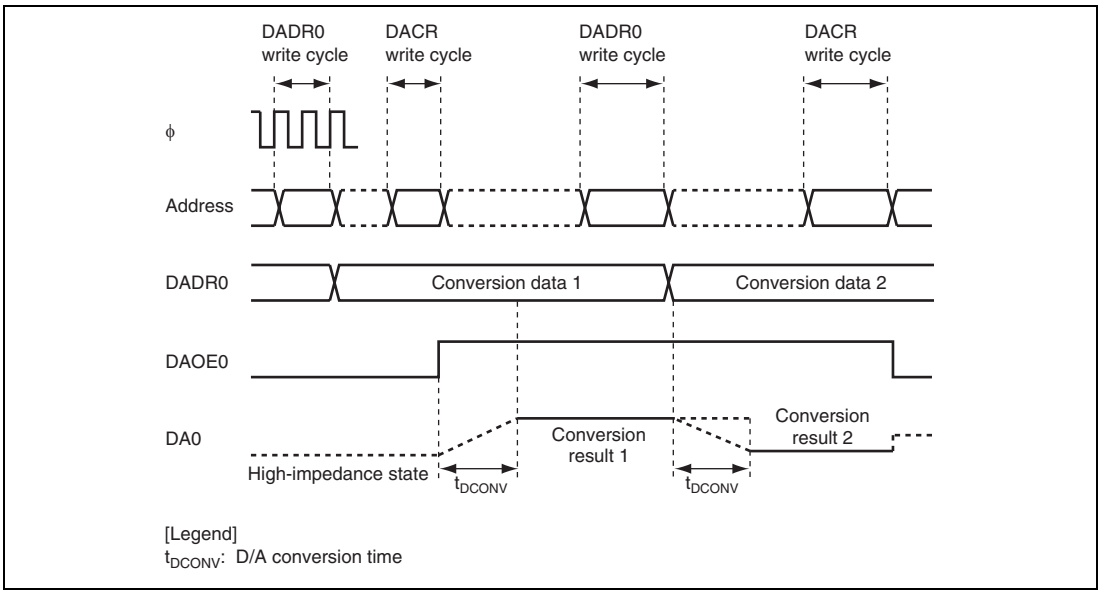


Figure 21.2 Example of D/A Converter Operation

21.5 Usage Notes

21.5.1 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 25, Power-Down Modes.

21.5.2 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.

21.5.3 D/A Conversion and D/A Output in Deep Standby Mode

When this LSI enters deep standby mode with D/A conversion enabled, the D/A conversion is stopped and thus the D/A outputs are also stopped. Before entering deep standby mode, clear the DAOE0, DAOE1, and DAE bits to 0 to disable the D/A outputs.

21.5.4 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 PVcc − 0.3 V ≤ AVcc ≤ PVcc and AVss = PVss. 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 (PVcc) and AVss to the ground (PVss).
2. Setting range of AVref input voltage
   Set the reference voltage range of the AVref pin as 3.0 V ≤ AVref ≤ AVcc.

Section 22 I/O Ports

This LSI has six ports: A to F.

All port pins are multiplexed with other pin functions. The functions of the multiplexed pins are selected using the pin function controller (PFC).

Each port is provided with a data register for storing the pin data and a port read register for reading out the pin values.

22.1 Port A

Port A is an I/O port with 32 pins shown in figure 22.1.

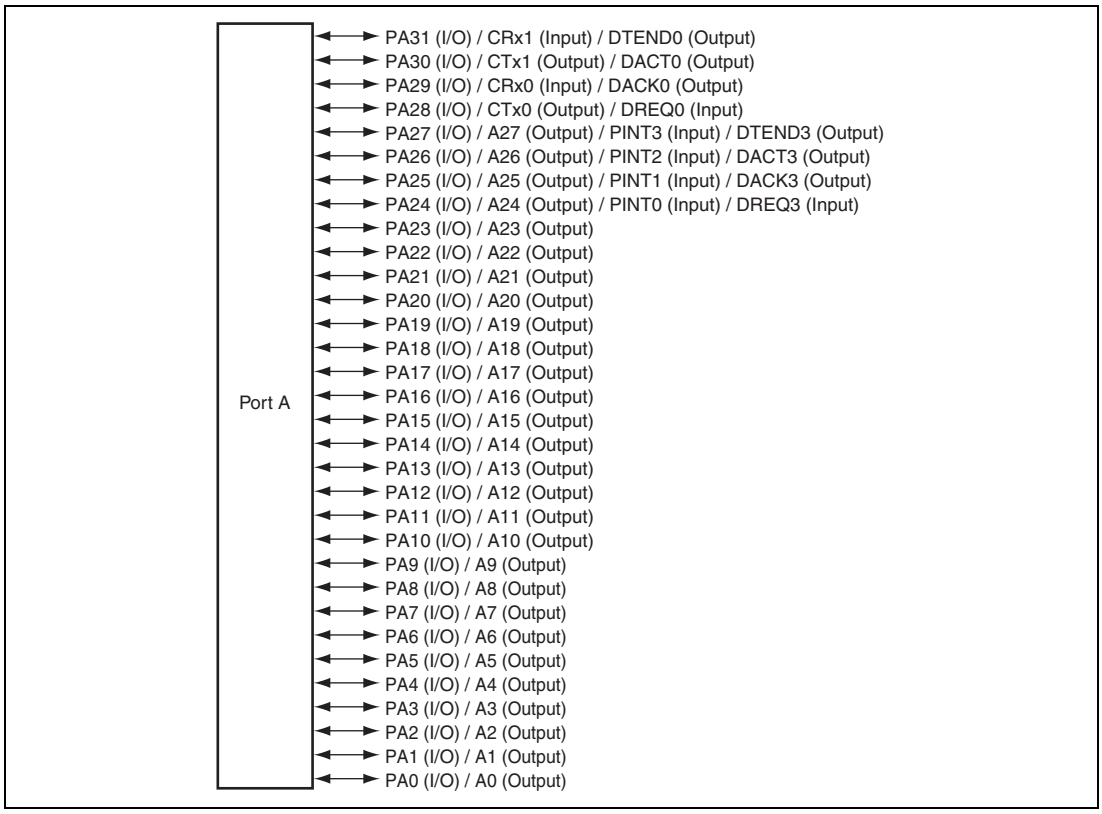


Figure 22.1 Port A

22.1.1 Register Configuration

Table 22.1 lists the port A registers.

Table 22.1 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port A data register H	PADRH	R/W	H'FFFE3800	8, 16, 32
Port A data register L	PADRL	R/W	H'FFFE3802	8, 16
Port A port register H	PAPRH	R	H'FFFE3804	8, 16, 32
Port A port register L	PAPRL	R	H'FFFE3806	8, 16

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

PADRH and PADRL are 16-bit readable/writable registers that store port A data. Bits PA31DR to PA0DR correspond to pins PA31 to PA0, respectively.

If a pin is set to the general output function, the pin will output the value written to the corresponding bit in PADRH or PADRL, and the register value is read from PADRH or PADRL regardless of the state of the pin.

If a pin is set to the general input function, the pin state, not the register value, will be returned if PADRH or PADRL is read. Also, if a value is written to PADRH or PADRL, although the value will actually be written, it will have no influence on the state of the pin. Table 22.2 summarizes the PADRH and PADRL read/write operations.

PADRH and PADRL are initialized to H'0000 by a power-on reset or in deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

- Port A Data Register H (PADRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA31 DR	PA30 DR	PA29 DR	PA28 DR	PA27 DR	PA26 DR	PA25 DR	PA24 DR	PA23 DR	PA22 DR	PA21 DR	PA20 DR	PA19 DR	PA18 DR	PA17 DR	PA16 DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- Port 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

Table 22.2 Port A Data Registers H and L (PADRH and PADRL) Read/Write Operations

PAIORH, PAIORL	Pin Function	Read	Write
0	General input	Pin state	The value is written to PADRH and PADRL but there is no effect on the pin state.
	Other than general input	Pin state	The value is written to PADRH and PADRL but there is no effect on the pin state.
1	General output	Value of PADRH and PADRL	The value written is output from the pin.
	Other than general output	Value of PADRH and PADRL	The value is written to PADRH and PADRL but there is no effect on the pin state.

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

PAPRH and PAPRL are 16-bit read-only registers in which bits PA31PR to PA0PR correspond to pins PA31 to PA0. PAPRH and PAPRL are always read as the states of the pins regardless of the PFC setting.

- Port A Port Register H (PAPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA31PR	PA30PR	PA29PR	PA28PR	PA27PR	PA26PR	PA25PR	PA24PR	PA23PR	PA22PR	PA21PR	PA20PR	PA19PR	PA18PR	PA17PR	PA16PR
Initial value:	PA31	PA30	PA29	PA28	PA27	PA26	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

- Port A Port Register L (PAPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
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

22.2 Port B

Port B is an I/O port with 32 pins shown in figure 22.2.

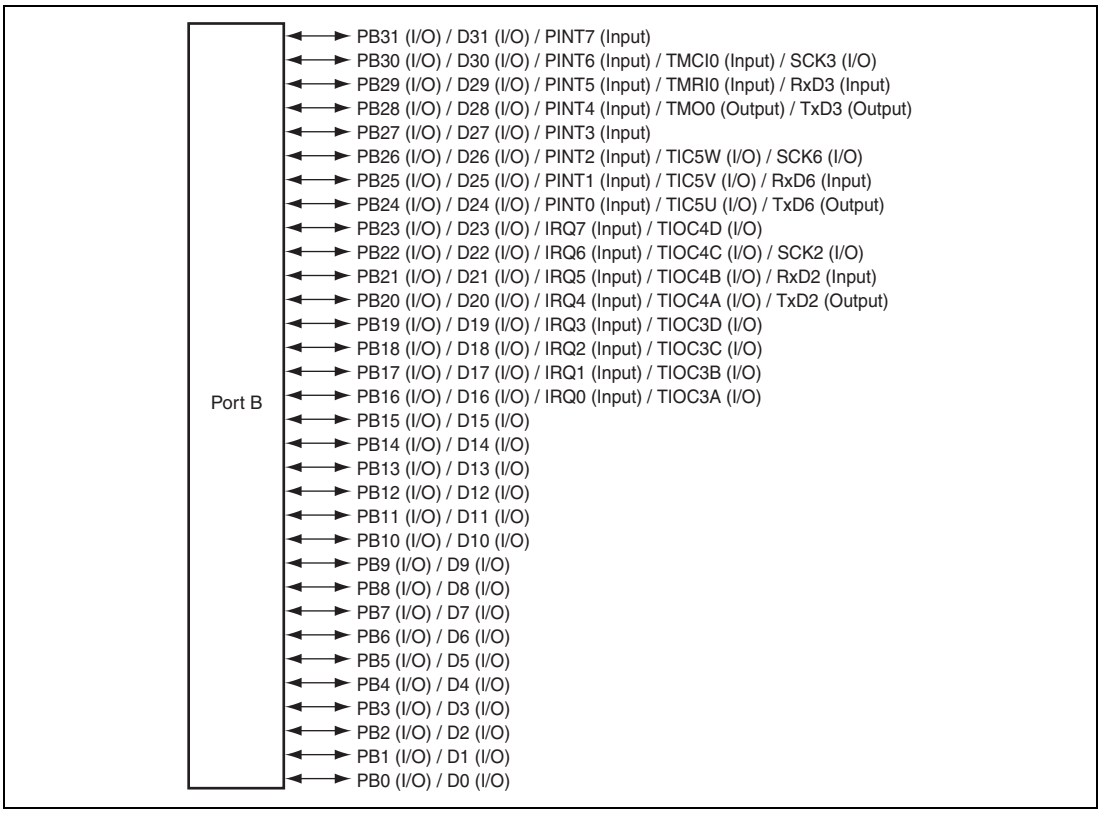


Figure 22.2 Port B

22.2.1 Register Configuration

Table 22.3 lists the port B registers.

Table 22.3 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port B data register H	PBDRH	R/W	H'FFFE3808	8, 16, 32
Port B data register L	PBDRL	R/W	H'FFFE380A	8, 16
Port B port register H	PBPRH	R	H'FFFE380C	8, 16, 32
Port B port register L	PBPRL	R	H'FFFE380E	8, 16

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

PBDRH and PBDRL are 16-bit readable/writable registers that store port B data. Bits PB31DR to PB0DR correspond to pins PB31 to PB0, respectively.

If a pin is set to the general output function, the pin will output the value written to the corresponding bit in PBDRH or PBDRL, and the register value is read from PBDRH or PBDRL regardless of the state of the pin.

If a pin is set to the general input function, the pin state, not the register value, will be returned if PBDRH or PBDRL is read. Also, if a value is written to PBDRH or PBDRL, although the value will actually be written, it will have no influence on the state of the pin. Table 22.4 summarizes the PBDRH and PBDRL read/write operations.

PBDRH and PBDRL are initialized to H'0000 by a power-on reset or in deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

- Port B data register H (PBDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB31 DR	PB30 DR	PB29 DR	PB28 DR	PB27 DR	PB26 DR	PB25 DR	PB24 DR	PB23 DR	PB22 DR	PB21 DR	PB20 DR	PB19 DR	PB18 DR	PB17 DR	PB16 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

- 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

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

PBIORH, L	Pin Function	Read	Write
0	General input	Pin state	The value is written to PBDRH and PBDRL but there is no effect on the pin state.
	Other than general input	Pin state	The value is written to PBDRH and PBDRL but there is no effect on the pin state.
1	General output	Value of PBDRH and PBDRL	The value written is output from the pin.
	Other than general output	Value of PBDRH and PBDRL	The value is written to PBDRH and PBDRL but there is no effect on the pin state.

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

PBPRH and PBPRL are 16-bit read-only registers in which bits PB31PR to PB0PR correspond to pins PB31 to PB0. PBPRH and PBPRL are always read the states of the pins regardless of the PFC setting.

- Port B Port Register H (PBPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB31PR	PB30PR	PB29PR	PB28PR	PB27PR	PB26PR	PB25PR	PB24PR	PB23PR	PB22PR	PB21PR	PB20PR	PB19PR	PB18PR	PB17PR	PB16PR
Initial value:	PB31	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

- Port B Port Register L (PBPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	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

22.3 Port C

Port C is an I/O port with 26 pins and is shown in figure 22.3.

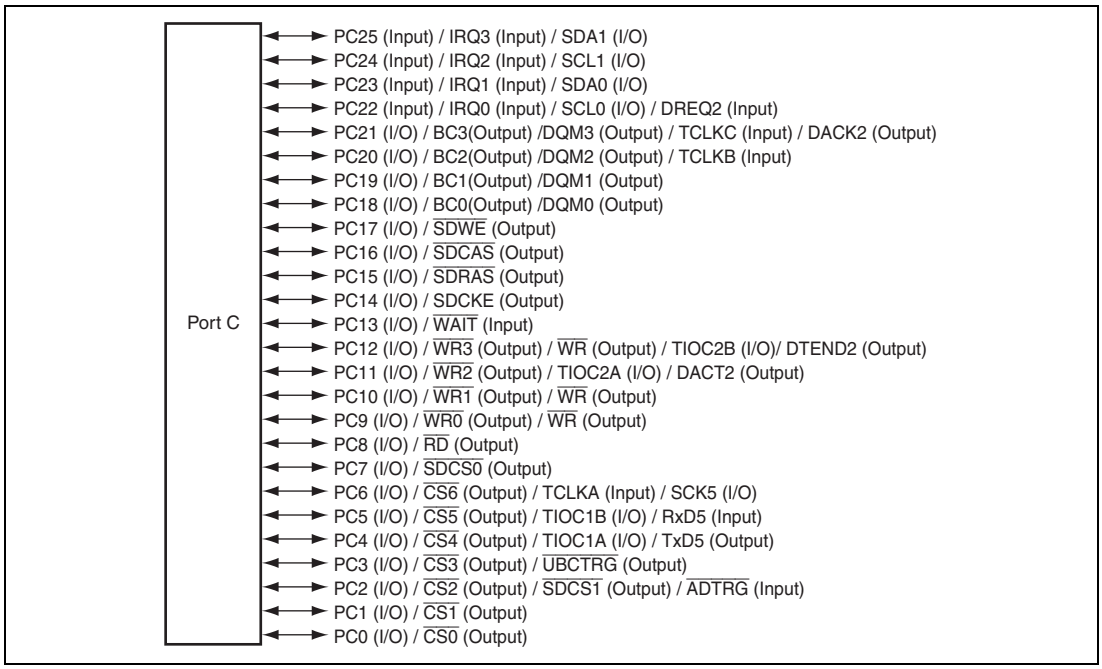


Figure 22.3 Port C

22.3.1 Register Configuration

Table 22.5 lists the port C registers.

Table 22.5 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port C data register H	PCDRH	R/W	H'FFFE3810	8, 16, 32
Port C data register L	PCDRL	R/W	H'FFFE3812	8, 16
Port C port register H	PCPRH	R	H'FFFE3814	8, 16, 32
Port C port register L	PCPRL	R	H'FFFE3816	8, 16

22.3.2 Port C Data Registers H and L (PCDRH and PCDRL)

PCDRH and PCDRL are 16-bit readable/writable registers that store port C data. Bits PC21DR to PC0DR correspond to pins PC21 to PC0, respectively.

If a pin is set to the general output function, the pin will output the value written to the corresponding bit in PCDRH or PCDRL, and the register value is read from PCDRH and PCDRL regardless of the state of the pin.

If a pin is set to the general input function, the pin state, not the register value, will be returned if PCDRH or PCDRL is read. Also, if a value is written to PCDRH or PCDRL, although the value will actually be written, it will have no influence on the state of the pin. Table 22.6 summarizes the PCDRH and PCDRL read/write operations.

Bits 15 to 6 in PCDRH are reserved. These bits are read as 0. The write value should always be 0.

PCDRH and PCDRL are initialized to H'0000 by a power-on reset or in deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

- Port C data register H (PCDRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PC21DR	PC20DR	PC19DR	PC18DR	PC17DR	PC16DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- Port C data register L (PCDRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.6 Port C Data Registers H and L (PCDRH and PCDRL) Read/Write Operations

PCIORH, PCIORL	Pin Function	Read	Write
0	General input	Pin state	The value is written to PCDRH and PCDRL but there is no effect on the pin state.
	Other than general input	Pin state	The value is written to PCDRH and PCDRL but there is no effect on the pin state.
1	General output	Value of PCDRH and PCDRL	The value written is output from the pin.
	Other than general output	Value of PCDRH and PCDRHL	The value is written to PCDRH and PCDRL but there is no effect on the pin state.

22.3.3 Port C Port Registers H and L (PCPRH and PCPRL)

PCPRH and PCPRL are 16-bit read-only registers in which bits PC25PR to PC0PR correspond to pins PC25 to PC0. PCPRH and PCPRL are always read as the states of the pins regardless of the PFC setting.

Bits 15 to 10 in PCPRH are reserved. These bits are read as 0. The write value should always be 0.

- Port C Port Register H (PCPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PC25PR	PC24PR	PC23PR	PC22PR	PC21PR	PC20PR	PC19PR	PC18PR	PC17PR	PC16PR
Initial value:	0	0	0	0	0	0	PC25	PC24	PC23	PC22	PC21	PC20	PC19	PC18	PC17	PC16
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

- Port C Port Register L (PCPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC5PR	PC3PR	PC2PR	PC1PR	PC0PR
Initial value:	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC5	PC3	PC2	PC1	PC0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

22.4 Port D

Port D is an I/O port with 17 pins shown in figure 22.4.

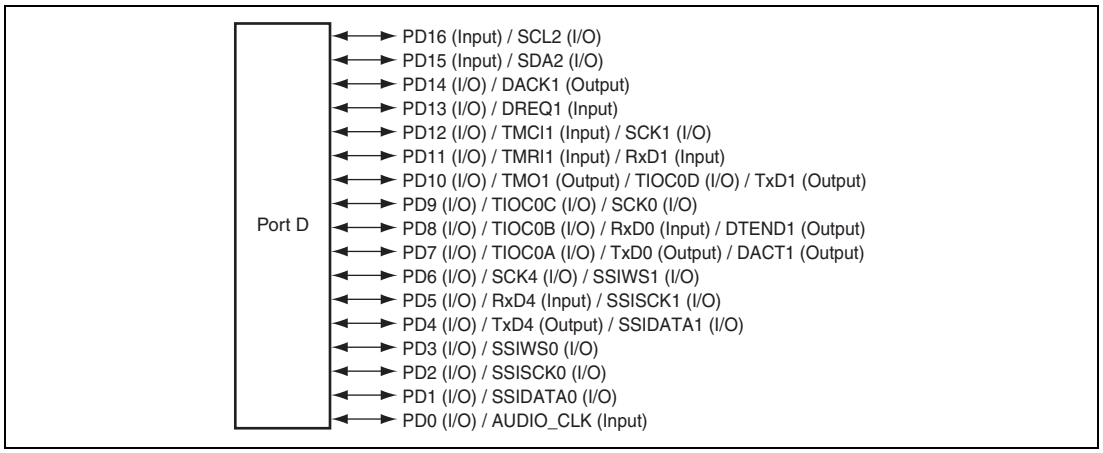


Figure 22.4 Port D

22.4.1 Register Configuration

Table 22.7 lists the port D registers.

Table 22.7 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port D data register	PDDR	R/W	H'FFFE381A	8, 16
Port D port register H	PDPRH	R	H'FFFE381C	8, 16, 32
Port D port register L	PDPRL	R	H'FFFE381E	8, 16

22.4.2 Port D Data Register (PDDR)

PDDR is a 16-bit readable/writable register that stores port D data. Bits PD14DR to PD0DR correspond to pins PD14 to PD0, respectively.

If a pin is set to the general output function, that pin will output the value written to the corresponding bit in PDDR, and the register value is read from PDDR regardless of the state of the pin.

If a pin is set to the general input function, the pin state, not the register value, will be returned if PDDR is read. Also, if a value is written to PDDR, although the value will actually be written, it will have no influence on the state of the pin. Table 22.8 summarizes the PDDR read/write operations.

Bit 15 in PDDR is reserved. This bit is read as 0. The write value should always be 0.

PDDR is initialized to H'0000 by a power-on reset or in deep standby mode. This register is not initialized either by a manual reset or by switching to sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
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

Table 22.8 Port D Data Register (PDDR) Read/Write Operations

PDIOR	Pin Function	Read	Write
0	General input	Pin state	The value is written to PDDR but there is no effect on the pin state.
	Other than general input	Pin state	The value is written to PDDR but there is no effect on the pin state.
1	General output	Value of PDDR	The value written is output from the pin.
	Other than general output	Value of PDDR	The value is written to PDDR but there is no effect on the pin state.

22.4.3 Port D Port Registers H and L (PDPRH and PDPRL)

PDPRH and PDPRL are 16-bit read-only registers in which bits PD16PR to PD0PR correspond to pins PD16 to PD0. PDPRH and PDPRL are always read as the states of the pins regardless of the PFC setting.

Bits 15 to 1 in PDPRH are reserved. These bits are read as 0. The write value should always be 0.

- Port D Port Register H (PDPRH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PD16PR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PD16
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

- Port D Port Register L (PDPRL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
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

22.5 Port E

Port E is an I/O port with 8 pins shown in figure 22.5.

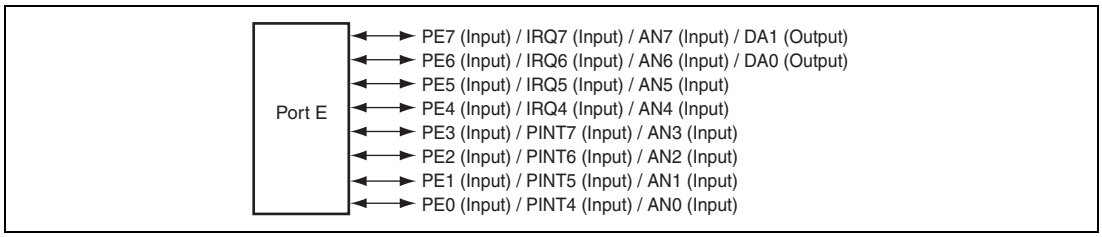


Figure 22.5 Port E

22.5.1 Register Configuration

Table 22.9 lists the port E registers.

Table 22.9 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port E port register	PEPR	R	H'FFFE3826	8, 16

22.5.2 Port E Port Register (PEPR)

PEPR is a 16-bit read-only register. Bits PE7PR to PE0PR correspond to pins PE7 to PE0, respectively. The pin values can always be read from PEPR, regardless of the PFC settings.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PE7 PR	PE6 PR	PE5 PR	PE4 PR	PE3 PR	PE2 PR	PE1 PR	PE0 PR
Initial value:	0	0	0	0	0	0	0	0	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

22.6 Port F

Port F is an I/O port with 8 pins shown in figure 22.6.

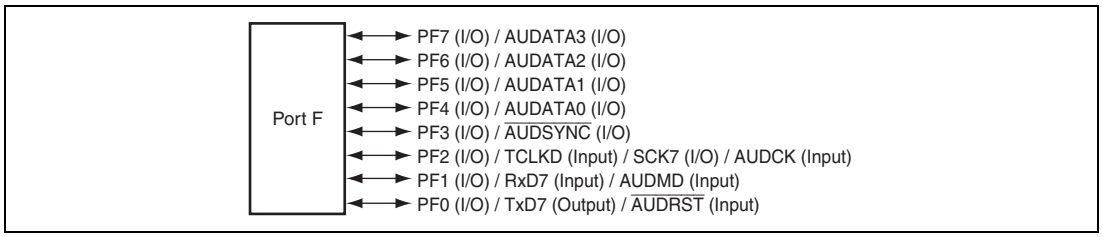


Figure 22.6 Port F

22.6.1 Register Configuration

Table 22.10 lists the port F registers.

Table 22.10 Register Configuration

Register Name	Abbreviation	R/W	Address	Access Size
Port F data register	PFDR	R/W	H'FFFE382A	8, 16
Port F port register	PFPR	R	H'FFFE382E	8, 16

22.6.2 Port F Data Register (PFDR)

PFDR is a 16-bit read-only register that stores the port F data. Bits PF7DR to PF0DR correspond to pins PF7 to PF0, respectively.

If a pin is set to the general output function, the pin will output the value written to the corresponding bit in PFDR, and the register value is read from PFDR regardless of the state of the pin.

If a pin is set to the general input function, the pin state, not the register value, will be returned if PFDR is read. Also, if a value is written to PFDR, although the value will actually be written, it will have no influence on the state of the pin. Table 22.11 summarizes the PFDR read/write operations.

PFDR is initialized to H'0000 by a power-on reset or in deep standby mode. This register is not initialized either by a manual reset or by switching to sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value:	0	0	0	0	0	0	0	0	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

Table 22.11 Port F Data Register (PFDR) Read/Write Operations

PFIOR	Pin Function	Read	Write
0	General input	Pin state	The value is written to PFDR but there is no effect on the pin state.
	Other than general input	Pin state	The value is written to PFDR but there is no effect on the pin state.
1	General output	Value of PFDR	The value written is output from the pin.
	Other than general output	Value of PFDR	The value is written to PFDR but there is no effect on the pin state.

22.6.3 Port F Port Register (PFPR)

PFPR is a 16-bit read-only register in which bits PF7PR to PF0PR correspond to pins PF7 to PF0. PFPR are always read as 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
	—	—	—	—	—	—	—	—	PF7 PR	PF6 PR	PF5 PR	PF4 PR	PF3 PR	PF2 PR	PF1 PR	PF0 PR
Initial value:	0	0	0	0	0	0	0	0	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Section 23 Pin Function Controller (PFC)

The pin function controller (PFC) consists of registers that select the functions of the multiplexed pins and their I/O directions. Tables 25.1 to 25.6 list the multiplexed pins of this LSI.

Table 23.1 Multiplexed Pin Table (Port A)

Port	Function 1 PAnMD[2:0] = 000 (Related modules)	Function 2 PAnMD[2:0] = 001 (Related modules)	Function 3 PAnMD[2:0] = 010 (Related modules)	Function 4 PAnMD[2:0] = 011 (Related modules)	Function 5 PAnMD[2:0] = 100 (Related modules)
A	PA31 I/O (Port)	CRx1 input (RCAN-ET)	DTEND0 output (DMAC)		
	PA30 I/O (Port)	CTx1 output (RCAN-ET)	DACT0 output (DMAC)		
	PA29 I/O (Port)	CRx0 input (RCAN-ET)	DACK0 output (DMAC)		
	PA28 I/O (Port)	CTx0 output (RCAN-ET)	DREQ0 input (DMAC)		
	PA27 I/O (Port)	A27 output (BSC)	DTEND3 output (DMAC)	PINT3B input (INTC)	
	PA26 I/O (Port)	A26 output (BSC)	DACT3 output (DMAC)	PINT2B input (INTC)	
	PA25 I/O (Port)	A25 output (BSC)	DACK3 output (DMAC)	PINT1B input (INTC)	
	PA24 I/O (Port)	A24 output (BSC)	DREQ3 input (DMAC)	PINT0B input (INTC)	
	PA23 I/O (Port)	A23 output (BSC)			
	PA22 I/O (Port)	A22 output (BSC)			
	PA21 I/O (Port)	A21 output (BSC)			
	PA20 I/O (Port)	A20 output (BSC)			
	PA19 I/O (Port)	A19 output (BSC)			
	PA18 I/O (Port)	A18 output (BSC)			
	PA17 I/O (Port)	A17 output (BSC)			
	PA16 I/O (Port)	A16 output (BSC)			
	PA15 I/O (Port)	A15 output (BSC)			
	PA14 I/O (Port)	A14 output (BSC)			

Port	Function 1 PAnMD[2:0] = 000 (Related modules)	Function 2 PAnMD[2:0] = 001 (Related modules)	Function 3 PAnMD[2:0] = 010 (Related modules)	Function 4 PAnMD[2:0] = 011 (Related modules)	Function 5 PAnMD[2:0] = 100 (Related modules)
A	PA13 I/O (Port)	A13 output (BSC)			
	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 23.2 Multiplexed Pin Table (Port B)

Port	Function 1 PBnMD[2:0] = 000 (Related modules)	Function 2 PBnMD[2:0] = 001 (Related modules)	Function 3 PBnMD[2:0] = 010 (Related modules)	Function 4 PBnMD[2:0] = 011 (Related modules)	Function 5 PBnMD[2:0] = 100 (Related modules)
B	PB31 I/O (Port)	D31 I/O (BSC)	PINT7A input (INTC)		
	PB30 I/O (Port)	D30 I/O (BSC)	PINT6A input (INTC)	SCK3 I/O (SCIF)	TMCI0 input (TMR)
	PB29 I/O (Port)	D29 I/O (BSC)	PINT5A input (INTC)	RxD3 input (SCIF)	TMRI0 input (TMR)
	PB28 I/O (Port)	D28 I/O (BSC)	PINT4A input (INTC)	TxD3 output (SCIF)	TMO0 output (TMR)
	PB27 I/O (Port)	D27 I/O (BSC)	PINT3A input (INTC)		
	PB26 I/O (Port)	D26 I/O (BSC)	PINT2A input (INTC)	TIC5W input (MTU2)	SCK6 I/O (SCIF)
	PB25 I/O (Port)	D25 I/O (BSC)	PINT1A input (INTC)	TIC5V input (MTU2)	RxD6 input (SCIF)
	PB24 I/O (Port)	D24 I/O (BSC)	PINT0A input (INTC)	TIC5U input (MTU2)	TxD6 output (SCIF)
	PB23 I/O (Port)	D23 I/O (BSC)	IRQ7A input (INTC)	TIOC4D I/O (MTU2)	
	PB22 I/O (Port)	D22 I/O (BSC)	IRQ6A input (INTC)	TIOC4C I/O (MTU2)	SCK2 I/O (SCIF)
	PB21 I/O (Port)	D21 I/O (BSC)	IRQ5A input (INTC)	TIOC4B I/O (MTU2)	RxD2 input (SCIF)
	PB20 I/O (Port)	D20 I/O (BSC)	IRQ4A input (INTC)	TIOC4A I/O (MTU2)	TxD2 output (SCIF)
	PB19 I/O (Port)	D19 I/O (BSC)	IRQ3A input (INTC)	TIOC3D I/O (MTU2)	
	PB18 I/O (Port)	D18 I/O (BSC)	IRQ2A input (INTC)	TIOC3C I/O (MTU2)	
	PB17 I/O (Port)	D17 I/O (BSC)	IRQ1A input (INTC)	TIOC3B I/O (MTU2)	
	PB16 I/O (Port)	D16 I/O (BSC)	IRQ0A input (INTC)	TIOC3A I/O (MTU2)	

Port	Function 1 PBnMD[2:0] = 000 (Related modules)	Function 2 PBnMD[2:0] = 001 (Related modules)	Function 3 PBnMD[2:0] = 010 (Related modules)	Function 4 PBnMD[2:0] = 011 (Related modules)	Function 5 PBnMD[2:0] = 100 (Related modules)
B	PB15 I/O (Port)	D15 I/O (BSC)			
	PB14 I/O (Port)	D14 I/O (BSC)			
	PB13 I/O (Port)	D13 I/O (BSC)			
	PB12 I/O (Port)	D12 I/O (BSC)			
	PB11 I/O (Port)	D11 I/O (BSC)			
	PB10 I/O (Port)	D10 I/O (BSC)			
	PB9 I/O (Port)	D9 I/O (BSC)			
	PB8 I/O (Port)	D8 I/O (BSC)			
	PB7 I/O (Port)	D7 I/O (BSC)			
	PB6 I/O (Port)	D6 I/O (BSC)			
	PB5 I/O (Port)	D5 I/O (BSC)			
	PB4 I/O (Port)	D4 I/O (BSC)			
	PB3 I/O (Port)	D3 I/O (BSC)			
	PB2 I/O (Port)	D2 I/O (BSC)			
	PB1 I/O (Port)	D1 I/O (BSC)			
	PB0 I/O (Port)	D0 I/O (BSC)			

Table 23.3 Multiplexed Pin Table (Port C)

Port	Function 1 PCnMD[2:0] = 000 (Related modules)	Function 2 PCnMD[2:0] = 001 (Related modules)	Function 3 PCnMD[2:0] = 010 (Related modules)	Function 4 PCnMD[2:0] = 011 (Related modules)	Function 5 PCnMD[2:0] = 100 (Related modules)
C	PC25 input (Port)	IRQ3B input (INTC)		SDA1 I/O (IIC3)	
	PC24 input (Port)	IRQ2B input (INTC)		SCL1 I/O (IIC3)	
	PC23 input (Port)	IRQ1B input (INTC)		SDA0 I/O (IIC3)	
	PC22 input (Port)	IRQ0B input (INTC)	DREQ2 input (DMAC)	SCL0 I/O (IIC3)	
	PC21 I/O (Port)	BC3/DQM3 output (BSC)	TCLKC input (MTU2)	DACK2 output (DMAC)	
	PC20 I/O (Port)	BC2/DQM2 output (BSC)	TCLKB input (MTU2)		
	PC19 I/O (Port)	BC1/DQM1 output (BSC)			
	PC18 I/O (Port)	BC0/DQM0 output (BSC)			
	PC17 I/O (Port)	$\overline{\text{SDWE}}$ output (BSC)			
	PC16 I/O (Port)	$\overline{\text{SDCAS}}$ output (BSC)			
	PC15 I/O (Port)	$\overline{\text{SDRAS}}$ output (BSC)			
	PC14 I/O (Port)	SDCKE output (BSC)			
	PC13 I/O (Port)	$\overline{\text{WAIT}}$ input (BSC)			
	PC12 I/O (Port)	$\overline{\text{WR3}}$ output (BSC)	TIOC2B I/O (MTU2)	DTEND2 output (DMAC)	
	PC11 I/O (Port)	$\overline{\text{WR2}}$ output (BSC)	TIOC2A I/O (MTU2)	DACT2 output (DMAC)	
	PC10 I/O (Port)	$\overline{\text{WR1}}$ output (BSC)			

Port	Function 1 PCnMD[2:0] = 000 (Related modules)	Function 2 PCnMD[2:0] = 001 (Related modules)	Function 3 PCnMD[2:0] = 010 (Related modules)	Function 4 PCnMD[2:0] = 011 (Related modules)	Function 5 PCnMD[2:0] = 100 (Related modules)
C	PC9 I/O (Port)	$\overline{WR0}$ output (BSC)			
	PC8 I/O (Port)	$\overline{RD}$ output (BSC)			
	PC7 I/O (Port)	$\overline{SDCS0}$ output (BSC)			
	PC6 I/O (Port)	$\overline{CS6}$ output (BSC)	SCK5 I/O (SCIF)	TCLKA input (MTU2)	
	PC5 I/O (Port)	$\overline{CS5}$ output (BSC)	RxD5 input (SCIF)	TIOC1B I/O (MTU2)	
	PC4 I/O (Port)	$\overline{CS4}$ output (BSC)	TxD5 output (SCIF)	TIOC1A I/O (MTU2)	
	PC3 I/O (Port)	$\overline{CS3}$ output (BSC)	$\overline{UBCTRG}$ output (UBC)		
	PC2 I/O (Port)	$\overline{CS2}$ output (BSC)	$\overline{SDCS1}$ output (BSC)	$\overline{ADTRG}$ input (A/D)	
	PC1 I/O (Port)	$\overline{CS1}$ output (BSC)			
	PC0 I/O (Port)	$\overline{CS0}$ output (BSC)			

Table 23.4 Multiplexed Pin Table (Port D)

Port	Function 1 PDnMD[2:0] = 000 (Related modules)	Function 2 PDnMD[2:0] = 001 (Related modules)	Function 3 PDnMD[2:0] = 010 (Related modules)	Function 4 PDnMD[2:0] = 011 (Related modules)	Function 5 PDnMD[2:0] = 100 (Related modules)
D	PD16 input (Port)	SCL2 I/O (IIC3)			
	PD15 input (Port)	SDA2 I/O (IIC3)			
	PD14 I/O (Port)		DACK1 output (DMAC)		
	PD13 I/O (Port)		DREQ1 input (DMAC)		
	PD12 I/O (Port)	SCK1 I/O (SCIF)	TMCI1 input (TMR)		
	PD11 I/O (Port)	RxD1 input (SCIF)	TMRI1 input (TMR)		
	PD10 I/O (Port)	TxD1 output (SCIF)	TMO1 output (TMR)	TIOC0D I/O (MTU2)	
	PD9 I/O (Port)	SCK0 I/O (SCIF)		TIOC0C I/O (MTU2)	
	PD8 I/O (Port)	RxD0 input (SCIF)	DTEND1 output (DMAC)	TIOC0B I/O (MTU2)	
	PD7 I/O (Port)	TxD0 output (SCIF)	DACT1 output (DMAC)	TIOC0A I/O (MTU2)	
	PD6 I/O (Port)	SSIWS1 I/O (SSI)	SCK4 I/O (SCIF)		
	PD5 I/O (Port)	SSISCK1 I/O (SSI)	RxD4 input (SCIF)		
	PD4 I/O (Port)	SSIDATA1 I/O (SSI)	TxD4 output (SCIF)		
	PD3 I/O (Port)	SSIWS0 I/O (SSI)			
	PD2 I/O (Port)	SSISCK0 I/O (SSI)			
	PD1 I/O (Port)	SSIDATA0 I/O (SSI)			
	PD0 I/O (Port)	AUDIO_CLK input (SSI)			

Table 23.5 Multiplexed Pin Table (Port E)

Port	Function 1 PEnMD[2:0] = 000 (Related modules)	Function 2 PEnMD[2:0] = 001 (Related modules)	Function 3 PEnMD[2:0] = 010 (Related modules)	Function 4 PEnMD[2:0] = 011 (Related modules)	Function 5 PEnMD[2:0] = 100 (Related modules)
E	PE7 input (Port)	IRQ7B input (INTC)			
	PE6 input (Port)	IRQ6B input (INTC)			
	PE5 input (Port)	IRQ5B input (INTC)			
	PE4 input (Port)	IRQ4B input (INTC)			
	PE3 input (Port)	PINT7B input (INTC)			
	PE2 input (Port)	PINT6B input (INTC)			
	PE1 input (Port)	PINT5B input (INTC)			
	PE0 input (Port)	PINT4B input (INTC)			

Table 23.6 Multiplexed Pin Table (Port F)

Port	Function 1 PFnMD[2:0] = 000 (Related modules)	Function 2 PFnMD[2:0] = 001 (Related modules)	Function 3 PFnMD[2:0] = 010 (Related modules)	Function 4 PFnMD[2:0] = 011 (Related modules)	Function 5 PFnMD[2:0] = 100 (Related modules)
F	PF7 I/O (Port)	AUDATA3 I/O (AUD-II)			
	PF6 I/O (Port)	AUDATA2 I/O (AUD-II)			
	PF5 I/O (Port)	AUDATA1 I/O (AUD-II)			
	PF4 I/O (Port)	AUDATA0 I/O (AUD-II)			
	PF3 I/O (Port)	$\overline{\text{AUDSYNC}}$ input (AUD-II)			
	PF2 I/O (Port)	AUDCK input (AUD-II)	SCK7 I/O (SCIF)	TCLKD input (MTU2)	
	PF1 I/O (Port)	AUDMD input (AUD-II)	RxD7 input (SCIF)		
	PF0 I/O (Port)	$\overline{\text{AUDRST}}$ input (AUD-II)	TxD7 output (SCIF)		

23.1 Register Descriptions

The PFC includes the following registers.

Table 23.7 Register Configuration

Register	Abbr.	R/W	Initial Value	Address	Access Size
Port A I/O register H	PAIORH	R/W	H'0000	H'FFFE3880	8, 16, 32
Port A I/O register L	PAIORL	R/W	H'0000	H'FFFE3882	8, 16
Port A control register 8	PACR8	R/W	H'0000	H'FFFE3884	8, 16, 32
Port A control register 7	PACR7	R/W	H'0000	H'FFFE3886	8, 16
Port A control register 6	PACR6	R/W	H'1111	H'FFFE3888	8, 16, 32
Port A control register 5	PACR5	R/W	H'1111	H'FFFE388A	8, 16
Port A control register 4	PACR4	R/W	H'1111	H'FFFE388C	8, 16, 32
Port A control register 3	PACR3	R/W	H'1111	H'FFFE388E	8, 16
Port A control register 2	PACR2	R/W	H'1111	H'FFFE3890	8, 16, 32
Port A control register 1	PACR1	R/W	H'1111	H'FFFE3892	8, 16
Port B I/O register H	PBIORH	R/W	H'0000	H'FFFE3898	8, 16, 32
Port B I/O register L	PBIORL	R/W	H'0000	H'FFFE389A	8, 16
Port B control register 8	PBCR8	R/W	H'0000/H'1111	H'FFFE389C	8, 16, 32
Port B control register 7	PBCR7	R/W	H'0000/H'1111	H'FFFE389E	8, 16
Port B control register 6	PBCR6	R/W	H'0000/H'1111	H'FFFE38A0	8, 16, 32
Port B control register 5	PBCR5	R/W	H'0000/H'1111	H'FFFE38A2	8, 16
Port B control register 4	PBCR4	R/W	H'0000/H'1111	H'FFFE38A4	8, 16, 32
Port B control register 3	PBCR3	R/W	H'0000/H'1111	H'FFFE38A6	8, 16
Port B control register 2	PBCR2	R/W	H'1111	H'FFFE38A8	8, 16, 32
Port B control register 1	PBCR1	R/W	H'1111	H'FFFE38AA	8, 16
Port C I/O register H	PCIORH	R/W	H'0000	H'FFFE38B0	8, 16, 32
Port C I/O register L	PCIORL	R/W	H'0000	H'FFFE38B2	8, 16
Port C control register 7	PCCR7	R/W	H'0000	H'FFFE38B6	8, 16
Port C control register 6	PCCR6	R/W	H'0000	H'FFFE38B8	8, 16, 32
Port C control register 5	PCCR5	R/W	H'0000	H'FFFE38BA	8, 16

Register	Abbr.	R/W	Initial Value	Address	Access Size
Port C control register 4	PCCR4	R/W	H'0000/H'0001	H'FFFE38BC	8, 16, 32
Port C control register 3	PCCR3	R/W	H'0011/H'0111/ H'1111	H'FFFE38BE	8, 16
Port C control register 2	PCCR2	R/W	H'0000	H'FFFE38C0	8, 16, 32
Port C control register 1	PCCR1	R/W	H'0001	H'FFFE38C2	8, 16
Port D I/O register	PDIOR	R/W	H'0000	H'FFFE38CA	8, 16
Port D control register 5	PDCR5	R/W	H'0000	H'FFFE38D2	8, 16
Port D control register 4	PDCR4	R/W	H'0000	H'FFFE38D4	8, 16, 32
Port D control register 3	PDCR3	R/W	H'0000	H'FFFE38D6	8, 16
Port D control register 2	PDCR2	R/W	H'0000	H'FFFE38D8	8, 16, 32
Port D control register 1	PDCR1	R/W	H'0000	H'FFFE38DA	8, 16
Port E control register 2	PECR2	R/W	H'0000	H'FFFE38F0	8, 16, 32
Port E control register 1	PECR1	R/W	H'0000	H'FFFE38F2	8, 16
Port F I/O register	PFIOR	R/W	H'0000	H'FFFE38FA	8, 16
Port F control register 2	PFCR2	R/W	H'0000	H'FFFE3908	8, 16, 32
Port F control register 1	PFCR1	R/W	H'0000	H'FFFE390A	8, 16

23.1.1 Port A I/O Registers H and L (PAIORH and PAIORL)

PAIORH and PAIORL are 16-bit readable/writable registers that select the I/O direction for the port A pins. Bits PA31IOR to PA0IOR correspond to pins PA31 to PA0, respectively. PAIORH and PAIORL are enabled when the function of the port A pins is set to general-purpose I/O (PA31 to PA0) by PACR, and are disabled in other cases. When a bit in PAIORH and PAIORL is set to 1, the corresponding pin is set to output, and when set to 0, the pin is set to input.

PAIORH and PAIORL are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to 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
	PA31 IOR	PA30 IOR	PA29 IOR	PA28 IOR	PA27 IOR	PA26 IOR	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/W	R/W	R/W	R/W	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 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

23.1.2 Port A Control Registers 1 to 8 (PACR1 to PACR8)

PACR1 to PACR8 are 16-bit readable/writable registers that select the functions of the multiplexed port A pins. When PINT3B to PINT0B are selected, do not set A input for the same interrupt.

PACR8 and PACR7 are initialized to H'0000 by a power-on reset or by switching to deep standby mode. PACR1 to PACR6 are initialized to H'1111 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

(1) Port A Control Register 8 (PACR8)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PA31MD[1:0]		—	—	PA30MD[1:0]		—	—	PA29MD[1:0]		—	—	PA28MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PA31MD [1:0]	00	R/W	PA31 Mode These bits control the function of the PA31/CRx1/ DTEND0 pin. 00: PA31 I/O (port) 01: CRx1 input (RCAN-ET) 10: DTEND0 output (DMAC) 11: Setting prohibited
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	PA30MD [1:0]	00	R/W	PA30 Mode These bits control the function of the PA30/CTx1/ DACT0 pin. 00: PA30 I/O (port) 01: CTx1 output (RCAN-ET) 10: DACT0 output (DMAC) 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PA29MD [1:0]	00	R/W	PA29 Mode These bits control the function of the PA29/CRx0/ DACK0 pin. 00: PA29 I/O (port) 01: CRx0 input (RCAN-ET) 10: DACK0 output (DMAC) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PA28MD [1:0]	00	R/W	PA28 Mode These bits control the function of the PA28/CTx0/ DREQ0 pin. 00: PA28 I/O (port) 01: CTx0 output (RCAN-ET) 10: DREQ0 input (DMAC) 11: Setting prohibited

(2) Port A Control Register 7 (PACR7)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA27MD[2:0]			—	PA26MD[2:0]			—	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/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	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	PA27MD [2:0]	000	R/W	PA27 Mode These bits control the function of the PA27/A27/ DTEND3/PINT3B pin. 000: PA27 I/O (port) 001: A27 output (BSC) 010: DTEND3 output (DMAC) 011: PINT3B 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.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PA26MD [2:0]	000	R/W	PA26 Mode These bits control the function of the PA26/A26/ DACT3/PINT2B pin. 000: PA26 I/O (port) 001: A26 output (BSC) 010: DACT3 output (DMAC) 011: PINT2B input (INTC) 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.
6 to 4	PA25MD [2:0]	000	R/W	PA25 Mode These bits control the function of the PA25/A25/ DACK3/PINT1B pin. 000: PA25 I/O (port) 001: A25 output (BSC) 010: DACK3 output (DMAC) 011: PINT1B input (INTC) 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PA24MD [1:0]	00	R/W	PA24 Mode These bits control the function of the PA24/A24/ DREQ3/PINT0B pin. 00: PA24 I/O (port) 01: A24 output (BSC) 10: DREQ3 input (DMAC) 11: PINT0B input (INTC)

(3) Port A Control Register 6 (PACR6)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PA23 MD0	—	—	—	PA22 MD0	—	—	—	PA21 MD0	—	—	—	PA20 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PA23MD0	1	R/W	PA23 Mode This bit controls the function of the PA23/A23 pin. 0: PA23 I/O (port) 1: A23 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA22MD0	1	R/W	PA22 Mode This bit controls the function of the PA22/A22 pin. 0: PA22 I/O (port) 1: A22 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA21MD0	1	R/W	PA21 Mode This bit controls the function of the PA21/A21 pin. 0: PA21 I/O (port) 1: A21 output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA20MD0	1	R/W	PA20 Mode This bit controls the function of the PA20/A20 pin. 0: PA20 I/O (port) 1: A20 output (BSC)

(4) Port A Control Register 5 (PACR5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PA19 MD0	—	—	—	PA18 MD0	—	—	—	PA17 MD0	—	—	—	PA16 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PA19MD0	1	R/W	PA19 Mode This bit controls the function of the PA19/A19 pin. 0: PA19 I/O (port) 1: A19 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA18MD0	1	R/W	PA18 Mode This bit controls the function of the PA18/A18 pin. 0: PA18 I/O (port) 1: A18 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA17MD0	1	R/W	PA17 Mode This bit controls the function of the PA17/A17 pin. 0: PA17 I/O (port) 1: A17 output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA16MD0	1	R/W	PA16 Mode This bit controls the function of the PA16/A16 pin. 0: PA16 I/O (port) 1: A16 output (BSC)

(5) Port A Control Register 4 (PACR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PA15 MD0	—	—	—	PA14 MD0	—	—	—	PA13 MD0	—	—	—	PA12 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PA15MD0	1	R/W	PA15 Mode This bit controls the function of the PA15/A15 pin. 0: PA15 I/O (port) 1: A15 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA14MD0	1	R/W	PA14 Mode This bit controls the function of the PA14/A14 pin. 0: PA14 I/O (port) 1: A14 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA13MD0	1	R/W	PA13 Mode This bit controls the function of the PA13/A13 pin. 0: PA13 I/O (port) 1: A13 output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA12MD0	1	R/W	PA12 Mode This bit controls the function of the PA12/A12 pin. 0: PA12 I/O (port) 1: A12 output (BSC)

(6) Port A Control Register 3 (PACR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PA11 MD0	—	—	—	PA10 MD0	—	—	—	PA9 MD0	—	—	—	PA8 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PA11MD0	1	R/W	PA11 Mode This bit controls the function of the PA11/A11 pin. 0: PA11 I/O (port) 1: A11 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA10MD0	1	R/W	PA10 Mode This bit controls the function of the PA10/A10 pin. 0: PA10 I/O (port) 1: A10 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA9MD0	1	R/W	PA9 Mode This bit controls the function of the PA9/A9 pin. 0: PA9 I/O (port) 1: A9 output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA8MD0	1	R/W	PA8 Mode This bit controls the function of the PA8/A8 pin. 0: PA8 I/O (port) 1: A8 output (BSC)

(7) Port A Control Register 2 (PACR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PA7 MD0	—	—	—	PA6 MD0	—	—	—	PA5 MD0	—	—	—	PA4 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PA7MD0	1	R/W	PA7 Mode This bit controls the function of the PA7/A7 pin. 0: PA7 I/O (port) 1: A7 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA6MD0	1	R/W	PA6 Mode This bit controls the function of the PA6/A6 pin. 0: PA6 I/O (port) 1: A6 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA5MD0	1	R/W	PA5 Mode This bit controls the function of the PA5/A5 pin. 0: PA5 I/O (port) 1: A5 output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA4MD0	1	R/W	PA4 Mode This bit controls the function of the PA4/A4 pin. 0: PA4 I/O (port) 1: A4 output (BSC)

(8) Port A Control Register 1 (PACR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PA3 MD0	—	—	—	PA2 MD0	—	—	—	PA1 MD0	—	—	—	PA0 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PA3MD0	1	R/W	PA3 Mode This bit controls the function of the PA3/A3 pin. 0: PA3 I/O (port) 1: A3 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PA2MD0	1	R/W	PA2 Mode This bit controls the function of the PA2/A2 pin. 0: PA2 I/O (port) 1: A2 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PA1MD0	1	R/W	PA1 Mode This bit controls the function of the PA1/A1 pin. 0: PA1 I/O (port) 1: A1 output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PA0MD0	1	R/W	PA0 Mode This bit controls the function of the PA0/A0 pin. 0: PA0 I/O (port) 1: A0 output (BSC)

23.1.3 Port B I/O Registers H and L (PBIORH and PBIORL)

PBIORH and PBIORL are 16-bit readable/writable registers that select the I/O direction for the port B pins. Bits PB31IOR to PB0IOR correspond to pins PB31 to PB0, respectively. PBIORH and PBIORL are enabled when the function of the port B pins is set to general-purpose I/O (PB31 to PB0) and to TIOC I/O (MTU2) by PBCR, and are disabled in other cases. When a bit in PBIORH and PBIORL is set to 1, the corresponding pin is set to output, and when set to 0, the pin is set to input.

PBIORH and PBIORL are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to 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
	PB31 IOR	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/W	R/W	R/W	R/W	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

23.1.4 Port B Control Registers 1 to 8 (PBCR1 to PBCR8)

PBCR1 to PBCR8 are 16-bit readable/writable registers that select the functions of the multiplexed port B pins. When IRQ7A to IRQ0A or PINT7A to PINT0A are selected, do not set B input for the same interrupt.

PBCR1 and PBCR2 are initialized to H'1111 by a power-on reset or by switching to deep standby mode. PBCR3 to PBCR8 are initialized to the values shown in table 23.8 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

Table 23.8 Port B Control Register Initial Values

	Initial Value		
Register	**Area 0: 32-Bit Mode**	**Area 0: 16-Bit Mode**	**Area 0: 8-Bit Mode**
PBCR5 to PBCR8	H'1111	H'0000	H'0000
PBCR3, PBCR4	H'1111	H'1111	H'0000

(1) Port B Control Register 8 (PBCR8)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PB31MD[1:0]		—	PB30MD[2:0]			—	PB29MD[2:0]			—	PB28MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value depends on the LSI's operating mode.

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	PB31MD [1:0]	00/01*	R/W	PB31 Mode These bits control the function of the PB31/D31/ PINT7A pin. 00: PB31 I/O (port) 01: D31 I/O (BSC) 10: PINT7A input (INTC) 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB30MD [2:0]	000/001*	R/W	PB30 Mode These bits control the function of the PB30/D30/ PINT6A/SCK3/TMCI0 pin. 000: PB30 I/O (port) 001: D30 I/O (BSC) 010: PINT6A input (INTC) 011: SCK3 I/O (SCIF) 100: TMCI0 input (TMR) 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	PB29MD [2:0]	000/001*	R/W	PB29 Mode These bits control the function of the PB29/D29/ PINT5A/RxD3/TMRI0 pin. 000: PB29 I/O (port) 001: D29 I/O (BSC) 010: PINT5A input (INTC) 011: RxD3 input (SCIF) 100: TMRI0 input (TMR) 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	PB28MD [2:0]	000/001*	R/W	PB28 Mode These bits control the function of the PB28/D28/ PINT4A/TxD3/TMO0 pin. 000: PB28 I/O (port) 001: D28 I/O (BSC) 010: PINT4A input (INTC) 011: TxD3 output (SCIF) 100: TMO0 output (TMR) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the LSI's operating mode.

(2) Port B Control Register 7 (PBCR7)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PB27MD[1:0]		—	PB26MD[2:0]			—	PB25MD[2:0]			—	PB24MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value depends on the LSI's operating mode.

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	PB27MD [1:0]	00/01*	R/W	PB27 Mode These bits control the function of the PB27/D27/ PINT3A pin. 00: PB27 I/O (port) 01: D27 I/O (BSC) 10: PINT3A input (INTC) 11: 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]	000/001*	R/W	PB26 Mode These bits control the function of the PB26/D26/ PINT2A/TIC5W/SCK6 pin. 000: PB26 I/O (port) 001: D26 I/O (BSC) 010: PINT2A input (INTC) 011: TIC5W input (MTU2) 100: SCK6 I/O (SCIF) 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	PB25MD [2:0]	000/001*	R/W	PB25 Mode These bits control the function of the PB25/D25/ PINT1A/TIC5V/RxD6 pin. 000: PB25 I/O (port) 001: D25 I/O (BSC) 010: PINT1A input (INTC) 011: TIC5V input (MTU2) 100: RxD6 input (SCIF) 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	PB24MD [2:0]	000/001*	R/W	PB24 Mode These bits control the function of the PB24/D24/ PINT0A/TIC5U/TxD6 pin. 000: PB24 I/O (port) 001: D24 I/O (BSC) 010: PINT0A input (INTC) 011: TIC5U input (MTU2) 100: TxD6 output (SCIF) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the LSI's operating mode.

(3) Port B Control Register 6 (PBCR6)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PB23MD[1:0]		—	PB22MD[2:0]			—	PB21MD[2:0]			—	PB20MD[2:0]		
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: * The initial value depends on the LSI's operating mode.

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	PB23MD [1:0]	00/01*	R/W	PB23 Mode These bits control the function of the PB23/D23/ IRQ7A/TIOC4D pin. 00: PB23 I/O (port) 01: D23 I/O (BSC) 10: IRQ7A input (INTC) 11: TIOC4D I/O (MTU2)
11	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
10 to 8	PB22MD [2:0]	000/001*	R/W	PB22 Mode These bits control the function of the PB22/D22/ IRQ6A/TIOC4C/SCK2 pin. 000: PB22 I/O (port) 001: D22 I/O (BSC) 010: IRQ6A input (INTC) 011: TIOC4C I/O (MTU2) 100: SCK2 I/O (SCIF) 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	PB21MD [2:0]	000/001*	R/W	PB21 Mode These bits control the function of the PB21/D21/ IRQ5A/TIOC4B/RxD2 pin. 000: PB21 I/O (port) 001: D21 I/O (BSC) 010: IRQ5A input (INTC) 011: TIOC4B I/O (MTU2) 100: RxD2 input (SCIF) 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	PB20MD [2:0]	000/001*	R/W	PB20 Mode These bits control the function of the PB20/D20/ IRQ4A/TIOC4A/TxD2 pin. 000: PB20 I/O (port) 001: D20 I/O (BSC) 010: IRQ4A input (INTC) 011: TIOC4A I/O (MTU2) 100: TxD2 output (SCIF) 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited

Note: * The initial value depends on the LSI's operating mode.

(4) Port B Control Register 5 (PBCR5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PB19MD[1:0]		—	—	PB18MD[2:0]		—	—	PB17MD[2:0]		—	—	PB16MD[2:0]	
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Note: * The initial value dependes on the LSI's clock operating mode.

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	PB19MD [1:0]	00/01*	R/W	PB19 Mode These bits control the function of the PB19/D19/ IRQ3A/TIOC3D pin. 00: PB19 I/O (port) 01: D19 I/O (BSC) 10: IRQ3A input (INTC) 11: TIOC3D I/O (MTU2)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PB18MD [1:0]	00/01*	R/W	PB18 Mode These bits control the function of the PB18/D18/ IRQ2A/TIOC3C pin. 00: PB18 I/O (port) 01: D18 I/O (BSC) 10: IRQ2A input (INTC) 11: TIOC3C I/O (MTU2)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PB17MD [1:0]	00/01*	R/W	PB17 Mode These bits control the function of the PB17/D17/ IRQ1A/TIOC3B pin. 00: PB17 I/O (port) 01: D17 I/O (BSC) 10: IRQ1A input (INTC) 11: TIOC3B I/O (MTU2)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PB16MD [1:0]	00/01*	R/W	PB16 Mode These bits control the function of the PB16/D16/ IRQ0A/TIOC3A pin. 00: PB16 I/O (port) 01: D16 I/O (BSC) 10: IRQ0A input (INTC) 11: TIOC3A I/O (MTU2)

Note: * The initial value depends on the LSI's operating mode.

(5) Port B Control Register 4 (PBCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PB15 MD0	—	—	—	PB14 MD0	—	—	—	PB13 MD0	—	—	—	PB12 MD0
Initial value:	0	0	0	0/1 *	0	0	0	0/1 *	0	0	0	0/1 *	0	0	0	0/1 *
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value dependes on the LSI's clock operating mode.

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	PB15MD0	0/1*	R/W	PB15 Mode This bit controls the function of the PB15/D15 pin. 0: PB15 I/O (port) 1: D15 I/O (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PB14MD0	0/1*	R/W	PB14 Mode This bit controls the function of the PB14/D14 pin. 0: PB14 I/O (port) 1: D14 I/O (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB13MD0	0/1*	R/W	PB13 Mode This bit controls the function of the PB13/D13 pin. 0: PB13 I/O (port) 1: D13 I/O (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PB12MD0	0/1*	R/W	PB12 Mode This bit controls the function of the PB12/D12 pin. 0: PB12 I/O (port) 1: D12 I/O (BSC)

Note: * The initial value depends on the LSI's operating mode.

(6) Port B Control Register 3 (PBCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PB11 MD0	—	—	—	PB10 MD0	—	—	—	PB9 MD0	—	—	—	PB8 MD0
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*	0	0	0	0/1*
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value dependes on the LSI's clock operating mode.

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	PB11MD0	0/1*	R/W	PB11 Mode This bit controls the function of the PB11/D11 pin. 0: PB11 I/O (port) 1: D11 I/O (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PB10MD0	0/1*	R/W	PB10 Mode This bit controls the function of the PB10/D10 pin. 0: PB10 I/O (port) 1: D10 I/O (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB9MD0	0/1*	R/W	PB9 Mode This bit controls the function of the PB9/D9 pin. 0: PB9 I/O (port) 1: D9 I/O (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PB8MD0	0/1*	R/W	PB8 Mode This bit controls the function of the PB8/D8 pin. 0: PB8 I/O (port) 1: D8 I/O (BSC)

Note: * The initial value depends on the LSI's operating mode.

(7) Port B Control Register 2 (PBCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PB7 MD0	—	—	—	PB6 MD0	—	—	—	PB5 MD0	—	—	—	PB4 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PB7MD0	1	R/W	PB7 Mode This bit controls the function of the PB7/D7 pin. 0: PB7 I/O (port) 1: D7 I/O (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PB6MD0	1	R/W	PB6 Mode This bit controls the function of the PB6/D6 pin. 0: PB6 I/O (port) 1: D6 I/O (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB5MD0	1	R/W	PB5 Mode This bit controls the function of the PB5/D5 pin. 0: PB5 I/O (port) 1: D5 I/O (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PB4MD0	1	R/W	PB4 Mode This bit controls the function of the PB4/D4 pin. 0: PB4 I/O (port) 1: D4 I/O (BSC)

(8) Port B Control Register 1 (PBCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PB3 MD0	—	—	—	PB2 MD0	—	—	—	PB1 MD0	—	—	—	PB0 MD0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PB3MD0	1	R/W	PB3 Mode This bit controls the function of the PB3/D3 pin. 0: PB3 I/O (port) 1: D3 I/O (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PB2MD0	1	R/W	PB2 Mode This bit controls the function of the PB2/D2 pin. 0: PB2 I/O (port) 1: D2 I/O (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PB1MD0	1	R/W	PB1 Mode This bit controls the function of the PB1/D1 pin. 0: PB1 I/O (port) 1: D1 I/O (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PB0MD0	1	R/W	PB0 Mode This bit controls the function of the PB0/D0 pin. 0: PB0 I/O (port) 1: D0 I/O (BSC)

23.1.5 Port C I/O Registers H and L (PCIORH and PCIORL)

PCIORH and PCIORL are 16-bit readable/writable registers that select the I/O direction for the port C pins. Bits PC21IOR to PC0IOR correspond to pins PC21 to PC0, respectively. PCIORH and PCIORL are enabled when the function of the port C pins is set to general-purpose I/O (PC21 to PC0) and to TIOC I/O (MTU2) by PCCR, and are disabled in other cases. When a bit in PCIORH and PCIORL is set to 1, the corresponding pin is set to output, and when set to 0, the pin is set to input.

Bits 15 to 6 in PCIORH are reserved. These bits are always read as 0. The write value should always be 0.

PCIORH and PCIORL are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

(1) Port C I/O Register H (PCIORH)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PC21 IOR	PC20 IOR	PC19 IOR	PC18 IOR	PC17 IOR	PC16 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

(2) Port C I/O Register L (PCIORL)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

23.1.6 Port C Control Registers 1 to 7 (PCCR1 to PCCR7)

PCCR1 to PCCR7 are 16-bit readable/writable registers that select the functions of the multiplexed port C pins. When IRQ3B to IRQ0B are selected, do not set A input for the same interrupt.

PCCR2, PCCR5, PCCR6, and PCCR7 are initialized to H'0000 by a power-on reset or by switching to deep standby mode. PCCR1 is initialized to H'0001 and PCCR3 and PCCR4 are initialized to the values shown in table 23.9 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

Table 23.9 Port C Control Register Initial Values

	Initial value		
Register	**Area 0: 32-Bit Mode**	**Area 0: 16-Bit Mode**	**Area 0: 8-Bit Mode**
PCCR4	H'0001	H'0000	H'0000
PCCR3	H'1111	H'0111	H'0011

(1) Port C Control Register 7 (PCCR7)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PC25MD1[1:0]		—	—	PC24MD1[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	Bit Name	Initial Value	R/W	Description
15 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PC25MD [1:0]	00	R/W	PC25 Mode These bits control the function of the PC25/IRQ3B/ SDA1 pin. 00: PC25 input (port) 01: IRQ3B input (INTC) 10: Setting prohibited 11: SDA1 I/O (IIC3)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PC24MD [1:0]	00	R/W	PC24 Mode These bits control the function of the PC24/IRQ2B/ SCL1 pin. 00: PC24 input (port) 01: IRQ2B input (INTC) 10: Setting prohibited 11: SCL1 I/O (IIC3)

(2) Port C Control Register 6 (PCCR6)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PC23MD[1:0]		—	—	PC22MD[1:0]		—	—	PC21MD[1:0]		—	—	PC20MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PC23MD [1:0]	00	R/W	PC23 Mode These bits control the function of the PC23/IRQ1B/ SDA0 pin. 00: PC23 input (port) 01: IRQ1B input (INTC) 10: Setting prohibited 11: SDA0 I/O (IIC3)
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	PC22MD [1:0]	00	R/W	PC22 Mode These bits control the function of the PC22/ IRQ0B/DREQ2/SCL0 pin. 00: PC22 input (port) 01: IRQ0B input (INTC) 10: DREQ2 input (DMAC) 11: SCL0 I/O (IIC3)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PC21MD [1:0]	00	R/W	PC21 Mode These bits control the function of the PC21/BC3/ DQM3/TCLKC/DACK2 pin. 00: PC21 input (port) 01: BC3/DQM3 output (BSC) 10: TCLKC input (MTU2) 11: DACK2 output (DMAC)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PC20MD [1:0]	00	R/W	PC20 Mode These bits control the function of the PC20/BC2/ DQM2/TCLKB pin. 00: PC20 I/O (port) 01: BC2/DQM2 output (BSC) 10: TCLKB input (MTU2) 11: Setting prohibited

(3) Port C Control Register 5 (PCCR5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PC19 MD0	—	—	—	PC18 MD0	—	—	—	PC17 MD0	—	—	—	PC16 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PC19MD0	0	R/W	PC19 Mode This bit controls the function of the PC19/ BC1/ DQM1 pin. 0: PC19 I/O (port) 1: BC1/DQM1 output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC18MD0	0	R/W	PC18 Mode This bit controls the function of the PC18/ BC0/ DQM0 pin. 0: PC18 I/O (port) 1: BC0/DQM0 output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC17MD0	0	R/W	PC17 Mode This bit controls the function of the PC17/$\overline{\text{SDWE}}$ pin. 0: PC17 I/O (port) 1: $\overline{\text{SDWE}}$ output (BSC)
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	PC16MD0	0	R/W	PC16 Mode This bit controls the function of the PC16/$\overline{\text{SDCAS}}$ pin. 0: PC16 I/O (port) 1: $\overline{\text{SDCAS}}$ output (BSC)

(4) Port C Control Register 4 (PCCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PC15MD0	—	—	—	PC14MD0	—	—	—	PC13MD0	—	—	PC12MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Note: * The initial value dependes on the LSI's clock operating mode.

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	PC15MD0	0	R/W	PC15 Mode This bit controls the function of the PC15/$\overline{\text{SDRAS}}$ pin. 0: PC15 I/O (port) 1: $\overline{\text{SDRAS}}$ output (BSC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC14MD0	0	R/W	PC14 Mode This bit controls the function of the PC14/SDCKE pin. 0: PC14 I/O (port) 1: SDCKE output (BSC)
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	PC13MD0	0	R/W	PC13 Mode This bit controls the function of the PC13/$\overline{\text{WAIT}}$ pin. 0: PC13 I/O (port) 1: $\overline{\text{WAIT}}$ input (BSC)
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PC12MD [1:0]	00/01*	R/W	PC12 Mode These bits control the function of the PC12/$\overline{\text{WR3}}$/ TIOC2B/DTEND2 pin. 00: PC12 I/O (port) 01: $\overline{\text{WR3}}$ output (BSC) 10: TIOC2B I/O (MTU2) 11: DTEND2 output (DMAC)

Note: * The initial value depends on the LSI's operating mode.

(5) Port C Control Register 3 (PCCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PC11MD[1:0]		—	—	—	PC10 MD0	—	—	—	PC9 MD0	—	—	—	PC8 MD0
Initial value:	0	0	0	0/1*	0	0	0	0/1*	0	0	0	1	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Note: * The initial value dependes on the LSI's clock operating mode.

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	PC11MD [1:0]	00/01*	R/W	PC11 Mode These bits control the function of the PC11/$\overline{\text{WR2}}$/ TIOC2A/DACT2 pin. 00: PC11 I/O (port) 01: $\overline{\text{WR2}}$ output (BSC) 10: TIOC2A I/O (MTU2) 11: DACT2 output (DMAC)

Bit	Bit Name	Initial Value	R/W	Description
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PC10MD0	0/1*	R/W	PC10 Mode This bit controls the function of the PC10/$\overline{\text{WR1}}$ pin. 0: PC10 I/O (port) 1: $\overline{\text{WR1}}$ output (BSC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC9MD0	1	R/W	PC9 Mode This bit controls the function of the PC9/$\overline{\text{WR0}}$ pin. 0: PC9 I/O (port) 1: $\overline{\text{WR0}}$ output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PC8MD0	1	R/W	PC8 Mode This bit controls the function of the PC8/$\overline{\text{RD}}$ pin. 0: PC8 I/O (port) 1: $\overline{\text{RD}}$ output (BSC)

Note: * The initial value depends on the LSI's operating mode.

(6) Port C Control Register 2 (PCCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PC7MD0	—	—	PC6MD[1:0]		—	—	PC5MD[1:0]		—	—	PC4MD[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	R	R	R/W	R/W	R	R	R/W	R/W	R	R	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.
12	PC7MD0	0	R/W	PC7 Mode This bit controls the function of the PC7/$\overline{\text{SDCS0}}$ pin. 0: PC7 I/O (port) 1: $\overline{\text{SDCS0}}$ output (BSC)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PC6MD[1:0]	00	R/W	PC6 Mode These bits control the function of the PC6/$\overline{\text{CS6}}$/SCK5/TCLKA pin. 00: PC6 I/O (port) 01: $\overline{\text{CS6}}$ output (BSC) 10: SCK5 I/O (SCIF) 11: TCLKA input (MTU2)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PC5MD[1:0]	00	R/W	PC5 Mode These bits control the function of the PC5/$\overline{\text{CS5}}$/RxD5/TIOC1B pin. 00: PC5 I/O (port) 01: $\overline{\text{CS5}}$ output (BSC) 10: RxD5 input (SCIF) 11: TICO1B I/O (MTU2)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PC4MD[1:0]	00	R/W	PC4 Mode These bits control the function of the PC4/$\overline{\text{CS4}}$/ TxD5/TIOC1A pin. 00: PC4 I/O (port) 01: $\overline{\text{CS4}}$ output (BSC) 10: TxD5 output (SCIF) 11: TIOC1A I/O (MTU2)

(7) Port C Control Register 1 (PCCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PC3MD[1:0]		—	—	PC2MD[1:0]		—	—	—	PC1 MD0	—	—	—	PC0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R	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, 12	PC3MD[1:0]	00	R/W	PC3 Mode These bits control the function of the PC3/$\overline{\text{CS3}}$/ $\overline{\text{UBCTRG}}$ pin. 00: PC3 I/O (port) 01: $\overline{\text{CS3}}$ output (BSC) 10: $\overline{\text{UBCTRG}}$ output (UBC) 11: Setting prohibited
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	PC2MD[1:0]	00	R/W	PC2 Mode These bits control the function of the PC2/$\overline{CS2}$/$\overline{SDCS1}$/$\overline{ADTRG}$ pin. 00: PC2 I/O (port) 01: $\overline{CS2}$ output (BSC) 10: $\overline{SDCS1}$ output (BSC) 11: $\overline{ADTRG}$ input (A/D)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PC1MD0	0	R/W	PC1 Mode This bit controls the function of the PC1/$\overline{CS1}$ pin. 0: PC1 I/O (port) 1: $\overline{CS1}$ output (BSC)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PC0MD0	1	R/W	PC0 Mode This bit controls the function of the PC0/$\overline{CS0}$ pin. 0: PC0 I/O (port) 1: $\overline{CS0}$ output (BSC)

23.1.7 Port D I/O Register (PDIOR)

PDIOR are 16-bit readable/writable registers that select the I/O direction for the port D pins. Bits PD14IOR to PD0IOR correspond to pins PD14 to PD0, respectively. PDIOR are enabled when the function of the port D pins is set to general-purpose I/O (PD14 to PD0) and to TIOC I/O (MTU2) by PDCR, and are disabled in other cases. When a bit in PDIOR is set to 1, the corresponding pin is set to output, and when set to 0, the pin is set to input.

Bits 15 in PDIOR is reserved. These bits are always read as 0. The write value should always be 0.

PDIOR are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
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

23.1.8 Port D Control Registers 1 to 5 (PDCR1 to PDCR5)

PDCR1 to PDCR5 are 16-bit readable/writable registers that select the functions of the multiplexed port D pins.

PDCR1 to PDCR5 are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

(1) Port D Control Register 5 (PDCR5)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PD16MD[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	PD16MD [1:0]	00	R/W	PD16 Mode These bits control the function of the PD16/SCL2 pin. 00: PD16 input (port) 01: SCL2 I/O (IIC3) 10: Setting prohibited 11: Setting prohibited

(2) Port D Control Register 4 (PDCR4)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PD15MD[1:0]		—	—	PD14MD[1:0]		—	—	PD13MD[1:0]		—	—	PD12MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PD15MD [1:0]	00	R/W	PD15 Mode These bits control the function of the PD15/ SDA2 pin. 00: PD15 input (port) 01: SDA2 I/O (IIC3) 10: Setting prohibited 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PD14MD [1:0]	00	R/W	PD14 Mode These bits control the function of the PD14/DACK1 pin. 00: PD14 I/O (port) 01: Setting prohibited 10: DACK1 output (DMAC) 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PD13MD [1:0]	000	R/W	PD13 Mode These bits control the function of the PD13/DREQ1 pin. 00: PD13 I/O (port) 01: Setting prohibited 10: DREQ1 input (DMAC) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PD12MD [1:0]	00	R/W	PD12 Mode These bits control the function of the PD12/ SCK1/TMCI1 pin. 00: PD12 I/O (port) 01: SCK1 I/O (SCIF) 10: TMCI1 input (TMR) 11: Setting prohibited

(3) Port D Control Register 3 (PDCR3)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PD11MD[1:0]		—	—	PD10MD[1:0]		—	—	PD9MD[1:0]		—	—	PD8MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PD11MD [1:0]	00	R/W	PD11 Mode These bits control the function of the PD11/ RxD1/TMRI1 pin. 00: PD11 I/O (port) 01: RxD1 input (SCIF) 10: TMRI1 output (TMR) 11: Setting prohibited
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PD10MD [1:0]	00	R/W	PD10 Mode These bits control the function of the PD10/ TxD1/TMO1/TIOC0D pin. 00: PD10 I/O (port) 01: TxD1 output (SCIF) 10: TMO1 output (TMR) 11: TIOC0D I/O (MTU2)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PD9MD[1:0]	00	R/W	PD9 Mode These bits control the function of the PD9/ SCK0/TIOC0C pin. 00: PD9 I/O (port) 01: SCK0 I/O (SCIF) 10: Setting prohibited 11: TIOC0C I/O (MTU2)

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PD8MD[1:0]	00	R/W	PD8 Mode These bits control the function of the PD8/RxD0/DTEND1/TIOC0B pin. 00: PD8 I/O (port) 01: RxD0 input (SCIF) 10: DTEND1 output (DMAC) 11: TIOC0B I/O (MTU2)

(4) Port D Control Register 2 (PDCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PD7MD[1:0]		—	—	PD6MD[1:0]		—	—	PD5MD[1:0]		—	—	PD4MD[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	R	R/W	R/W	R	R	R/W	R/W	R	R	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, 12	PD7MD[1:0]	00	R/W	PD7 Mode These bits control the function of the PD7/TxD0/DACT1/TIOC0A pin. 00: PD7 I/O (port) 01: TxD0 output (SCIF) 10: DACT1 output (DMAC) 11: TIOC0A I/O (MTU2)
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	PD6MD[1:0]	00	R/W	PD6 Mode These bits control the function of the PD6/ SSIWS1/SCK4 pin. 00: PD6 I/O (port) 01: SSIWS1 I/O (SSI) 10: SCK4 I/O (SCIF) 11: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PD5MD[1:0]	00	R/W	PD5 Mode These bits control the function of the PD5/ SSICK1/RxD4 pin. 00: PD5 I/O (port) 01: SSISCK1 I/O (SSI) 10: RxD4 input (SCIF) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PD4MD[1:0]	00	R/W	PD4 Mode These bits control the function of the PD4/ SSIDATA1/TxD4 pin. 00: PD4 I/O (port) 01: SSIDATA1 I/O (SSI) 10: TxD4 output (SCIF) 11: Setting prohibited

(5) Port D Control Register 1 (PDCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PD3 MD0	—	—	—	PD2 MD0	—	—	—	PD1 MD0	—	—	—	PD0 MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PD3MD0	0	R/W	PD3 Mode This bit controls the function of the PD3/SSIWS0 pin. 0: PD3 I/O (port) 1: SSIWS0 I/O (SSI)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PD2MD0	0	R/W	PD2 Mode This bit controls the function of the PD2/SICK0 pin. 0: PD2 I/O (port) 1: SSISCK0 I/O (SSI)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PD1MD0	0	R/W	PD1 Mode This bit controls the function of the PD1/SSIDATA0 pin. 0: PD1 I/O (port) 1: SSIDATA0 I/O (SSI)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PD0MD0	0	R/W	PD0 Mode This bit controls the function of the PD0/ AUDIO_CLK pin. 0: PD0 I/O (port) 1: AUDIO_CLK input (SSI)

23.1.9 Port E Control Registers 1 and 2 (PECR1 and PECR2)

PECR1 and PECR2 are 16-bit readable/writable registers that select the functions of the multiplexed port E pins. The pins states are set by the corresponding module for A/D converter input and for D/A converter output. When IRQ7B to IRQ4B or PINT7B to PINT4B are selected, do not set A input for the same interrupt.

PECR1 and PECR2 are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

(1) Port E Control Register 2 (PECR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PE7MD0	—	—	—	PE6MD0	—	—	—	PE5MD0	—	—	—	PE4MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PE7MD0	0	R/W	PE7 Mode This bit controls the function of the PE7/IRQ7B pin. 0: PE7 input (port) 1: IRQ7B input (INTC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PE6MD0	0	R/W	PE6 Mode This bit controls the function of the PE6/IRQ6B pin. 0: PE6 input (port) 1: IRQ6B input (INTC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PE5MD0	0	R/W	PE5 Mode This bit controls the function of the PE5/IRQ5B pin. 0: PE5 input (port) 1: IRQ5B input (INTC)

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PE4MD0	0	R/W	PE4 Mode This bit controls the function of the PE4/IRQ4B pin. 0: PE4 input (port) 1: IRQ4B input (INTC)

(2) Port E Control Register 1 (PECR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PE3MD0	—	—	—	PE2MD0	—	—	—	PE1MD0	—	—	—	PE0MD0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	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.
12	PE3MD0	0	R/W	PE3 Mode This bit controls the function of the PE3/PINT7B pin. 0: PE3 input (port) 1: PINT7B input (INTC)
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	PE2MD0	0	R/W	PE2 Mode This bit controls the function of the PE2/PINT6B pin. 0: PE2 input (port) 1: PINT6B input (INTC)
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	PE1MD0	0	R/W	PE1 Mode This bit controls the function of the PE1/PINT5B pin. 0: PE1 input (port) 1: PINT5B input (INTC)

Bit	Bit Name	Initial Value	R/W	Description
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PE0MD0	0	R/W	PE0 Mode This bit controls the function of the PE0/PINT4B pin. 0: PE0 input (port) 1: PINT4B input (INTC)

23.1.10 Port F I/O Register (PFIOR)

PFIOR is a 16-bit readable/writable register that selects the I/O direction for the port F pins. Bits PF7IOR to PF0IOR correspond to pins PF7 to PF0, respectively. PFIOR is enabled when the function of the port F pins is set to general-purpose I/O (PF7 to PF0) by PFCR, and are disabled in other cases. When a bit in PFIOR is set to 1, the corresponding pin is set to output, and when set to 0, the pin is set to input.

Bits 15 to 8 in PFIOR are reserved. These bits are always read as 0. The write value should always be 0.

PFIOR is initialized to H'0000 by a power-on reset or by switching to deep standby mode. This register is not initialized either by a manual reset or by switching to sleep mode or software standby mode.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
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

23.1.11 Port F Control Registers 1 and 2 (PFCR1 and PFCR2)

PFCR1 and PFCR2 are 16-bit readable/writable registers that select the functions of the multiplexed port F pins.

PFCR1 and PFCR2 are initialized to H'0000 by a power-on reset or by switching to deep standby mode. These registers are not initialized either by a manual reset or by switching to sleep mode or software standby mode.

(1) Port F Control Register 2 (PFCR2)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PF7MD0	—	—	PF6MD[1:0]		—	—	PF5MD[1:0]		—	—	PF4MD[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	R	R	R/W	R/W	R	R	R/W	R/W	R	R	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.
12	PF7MD0	0	R/W	PF7 Mode This bit controls the function of the PF7/AUDATA3 pin. 0: PF7 I/O (port) 1: AUDATA3 I/O (AUD-II)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF6MD[1:0]	00	R/W	PF6 Mode These bits control the function of the PF6/ AUDATA2 pin. 00: PF6 I/O (port) 01: AUDATA2 I/O (AUD-II) 10: Setting prohibited 11: Setting prohibited

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, 4	PF5MD[1:0]	00	R/W	PF5 Mode These bits control the function of the PF5/ AUDATA1 pin. 00: PF5 I/O (port) 01: AUDATA1 I/O (AUD-II) 10: Setting prohibited 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF4MD[1:0]	00	R/W	PF4 Mode These bits control the function of the PF4/ AUDATA0 pin. 00: PF4 I/O (port) 01: AUDATA0 I/O (AUD-II) 10: Setting prohibited 11: Setting prohibited

(2) Port F Control Register 1 (PFCR1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PF3MD0	—	—	PF2MD[1:0]		—	—	PF1MD[1:0]		—	—	PF0MD[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	R	R	R/W	R/W	R	R	R/W	R/W	R	R	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.
12	PF3MD0	0	R/W	PF3 Mode This bit controls the function of the PF3/ $\overline{\text{AUDSYNC}}$ pin. 0: PF3 I/O (port) 1: $\overline{\text{AUDSYNC}}$ input (AUD-II)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PF2MD[1:0]	00	R/W	PF2 Mode These bits control the function of the PF2/ AUDCK/SCK7/TCLKD pin. 00: PF2 I/O (port) 01: AUDCK input (AUD-II) 10: SCK7 I/O (SCIF) 11: TCLKD input (MTU2)
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
5, 4	PF1MD[1:0]	00	R/W	PF1 Mode These bits control the function of the PF1/ AUDMD/RxD7 pin. 00: PF1 I/O (port) 01: AUDMD input (AUD-II) 10: RxD7 input (SCIF) 11: Setting prohibited
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PF0MD[1:0]	00	R/W	PF0 Mode These bits control the function of the PF0/ $\overline{\text{AUDRST}}$/TxD7 pin. 00: PF0 I/O (port) 01: $\overline{\text{AUDRST}}$ input (AUD-II) 10: TxD7 output (SCIF) 11: Setting prohibited

23.2 Usage Note

The settings of the port control registers are used as the output pin select signals, and are not basically used as the input pin select signals. This causes the signals input from the pins to propagate to all the modules having the relevant multiplexed pins. So, unnecessary input signals must be disabled by the settings of the respective modules.

Settings of port control registers are decoded to enable/disable pins IRQ7A to IRQ0A and IRQ7B to IRQ0B or pins PINT7A to PINT0A and PINT7B to PINT0B. Be sure to select either one of them.

Section 24 On-Chip RAM

This LSI has an on-chip RAM module that achieves high-speed access and can 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.

24.1 Features

- Pages
  Two pages (pages 0 and 1) are provided.
- Memory map
  The on-chip RAM is located in the address spaces shown in table 24.1.

Table 24.1 On-Chip RAM Address Spaces

Page	Address
Page 0	H'FFF80000 to H'FFF83FFF
Page 1	H'FFF84000 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 via the internal DMA write bus/internal DMA read bus and bus bridge.
- Priority
  When requests for access to the same page from different buses coincide, the access is processed in priority order. The priority is I bus > M bus > F bus.

24.2 Usage Notes

24.2.1 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 modules or different pages are accessed by each bus.

24.2.2 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 RAM page 0
  MOV.L  #H'FFF80000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0

// For RAM page 1
  MOV.L  #H'FFF84000,R0
  MOV.L  @R0,R1
  MOV.L  R1,@R0
```

Figure 24.1 Examples of Read/Write before Disabling RAM

Section 25 Power-Down Modes

This LSI supports sleep mode, software standby mode, deep standby mode, and module standby mode. In power-down modes, functions of CPU, clocks, on-chip memory, or part of on-chip peripheral modules are halted or the power-supply is turned off, through which low power consumption is achieved. These modes are canceled by a reset or interrupt.

25.1 Features

25.1.1 Power-Down Modes

This LSI has the following power-down modes and function:

1. Sleep mode
2. Software standby mode
3. Deep standby mode
4. Module standby function

Table 25.1 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 25.1 States of Power-Down Modes

Power-Down Mode	Transition Conditions	State*1								Canceling Procedure
		CPG	CPU	CPU Register	On-Chip RAM	On-Chip Peripheral Modules	RTC	Power supply	External Memory	
Sleep mode	Execute SLEEP instruction with STBY bit in STBCR cleared to 0	Runs	Halts	Held	Runs	Runs	Runs*2	Runs	Auto-refreshing	• Interrupt • Manual reset • Power-on reset • Bus error
Software standby mode	Execute SLEEP instruction with STBY bit in STBCR set to 1 and DEEP bit to 0	Halts	Halts	Held	Halts (contents are held)	Halts	Runs*2	Runs	Self-refreshing	• NMI interrupt • IRQ interrupt • Manual reset • Power-on reset
Deep standby mode	Execute SLEEP instruction with STBY and DEEP bits in STBCR set to 1	Halts	Halts	Halts	Halts (contents are held*3)	Halts	Runs*2	Halts	Self-refreshing	• NMI interrupt*4 • IRQ interrupt*4 (only for PE7 to PE4 and PC25 to PC22) • Manual reset*4 • Power-on reset*4
Module standby function	Set the MSTP bits in STBCR2 to STBCR5 to 1	Runs	Runs	Held	Runs	Specified module halts	Halts	Runs	Auto-refreshing	• Clear MSTP bit to 0 • Power-on reset (only for RTC, H-UDI, UBC, DMAC, and AUD-II)

Notes: 1. The pin state is retained or set to high impedance. For details, see appendix A, Pin States.

2. RTC operates when the START bit in the RCR2 register is set to 1. For details, see section 15, Realtime Clock (RTC).

3. Setting bits RAMKP3 to RAMKP0 in the RAMKP register to 1 enables the retention of data in the corresponding area in the on-chip RAM during the transition to deep standby mode. However, when deep standby mode is canceled by a power-on reset, the contents in the corresponding on-chip RAM area are not retained.

4. Deep standby mode can be canceled by an interrupt (NMI or IRQ) or a reset (manual reset or power-on reset). However, IRQ is reset only by PE7 to PE0 and PC25 to PC22. When deep standby mode is canceled by NMI interrupt or IRQ interrupt, reset exception handling is executed instead of interrupt exception handling. These are power-on reset exception handlings including a manual reset.

25.2 Register Descriptions

The following registers are used in power-down modes.

Table 25.2 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'1E	H'FFFE0018	8
Standby control register 3	STBCR3	R/W	H'3F	H'FFFE0408	8
Standby control register 4	STBCR4	R/W	H'FF	H'FFFE040C	8
Standby control register 5	STBCR5	R/W	H'FF	H'FFFE0410	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
RAM retaining area specifying register	RAMKP	R/W	H'00	H'FFFF1907	8
Deep standby oscillation stabilizing clock select register	DSCNT	R/W	H'00	H'FFFF1906	8
Deep standby cancel source flag register	DSFR	R/W	H'0000	H'FFFF1904	16

25.2.1 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 or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	STBY	DEEP	—	—	—	—	MSTP1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
7	STBY	0	R/W	Software Standby, Deep Standby
6	DEEP	0	R/W	Specifies transition to software standby mode or deep standby mode. 0x: Executing SLEEP instruction puts chip into sleep mode. 10: Executing SLEEP instruction puts chip into software standby mode. 11: Executing SLEEP instruction puts chip into deep standby mode.
5 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	MSTP1	0	R/W	Module Stop 1 Setting the MSTP1 bit to 1 stops supplying clock to RTC 0: RTC runs 1: Stops supplying clock to RTC
0	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

[Legend]
x: Don't care

25.2.2 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'1E by a power-on reset or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	MSTP10	MSTP9	MSTP8	—	MSTP6	MSTP5	—	MSTP3
Initial value:	0	0	0	1	1	1	1	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	MSTP6	1	R/W	Module Stop 6 When the MSTP6 bit is set to 1, the supply of the clock to the RCAN-ET0 is halted. 0: RCAN-ET0 runs. 1: Clock supply to RCAN-ET0 halted.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP5	1	R/W	Module Stop 5 When the MSTP5 bit is set to 1, the supply of the clock to the RCAN-ET1 is halted. 0: RCAN-ET1 runs. 1: Clock supply to RCAN-ET1 halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
0	MSTP3	0	R/W	Module Stop 3 When the MSTP3 bit is set to 1, the supply of the clock to the AUD-II is halted. 0: AUD-II runs. 1: Clock supply to AUD-II halted.

25.2.3 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'3F by a power-on reset or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	—	—	MSTP35	—	MSTP33	MSTP32	MSTP31	—
Initial value:	0	0	1	1	1	1	1	1
R/W:	R	R	R/W	R	R/W	R/W	R/W	R

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	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.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
3	MSTP33	1	R/W	Module Stop 33 When the MSTP33 bit is set to 1, the supply of the clock to the TMR is halted. 0: TMR runs. 1: Clock supply to TMR 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.

Bit	Bit Name	Initial Value	R/W	Description
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	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.

25.2.4 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'FF by a power-on reset or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	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.

Bit	Bit Name	Initial Value	R/W	Description
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	MSTP43	1	R/W	Module Stop 43 When the MSTP43 bit is set to 1, the supply of the clock to the SCIF4 is halted. 0: SCIF4 runs. 1: Clock supply to SCIF4 halted.
2	MSTP42	1	R/W	Module Stop 42 When the MSTP42 bit is set to 1, the supply of the clock to the SCIF5 is halted. 0: SCIF5 runs. 1: Clock supply to SCIF5 halted.
1	MSTP41	1	R/W	Module Stop 41 When the MSTP41 bit is set to 1, the supply of the clock to the SCIF6 is halted. 0: SCIF6 runs. 1: Clock supply to SCIF6 halted.
0	MSTP40	1	R/W	Module Stop 40 When the MSTP40 bit is set to 1, the supply of the clock to the SCIF7 is halted. 0: SCIF7 runs. 1: Clock supply to SCIF7 halted.

25.2.5 Standby Control Register 5 (STBCR5)

STBCR5 is an 8-bit readable/writable register that controls the operation of modules in power-down modes. STBCR5 is initialized to H'FF by a power-on reset or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	MSTP55	—	MSTP53	MSTP52	—	CKDV3
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	MSTP57	1	R/W	Module Stop 57 When the MSTP57 bit is set to 1, the supply of the clock to the IIC30 is halted. 0: IIC30 runs. 1: Clock supply to IIC30 halted.
6	MSTP56	1	R/W	Module Stop 56 When the MSTP56 bit is set to 1, the supply of the clock to the IIC31 is halted. 0: IIC31 runs. 1: Clock supply to IIC31 halted.
5	MSTP55	1	R/W	Module Stop 55 When the MSTP55 bit is set to 1, the supply of the clock to the IIC32 is halted. 0: IIC32 runs. 1: Clock supply to IIC32 halted.
4	—	1	R	Reserved This bit is always read as 1. The write value should always be 0.
3	MSTP53	1	R/W	Module Stop 53 When the MSTP53 bit is set to 1, the supply of the clock to the SSI0 is halted. 0: SSI0 runs. 1: Clock supply to SSI0 halted.

Bit	Bit Name	Initial Value	R/W	Description
2	MSTP52	1	R/W	Module Stop 52 When the MSTP52 bit is set to 1, the supply of the clock to the SSI1 is halted. 0: SSI1 runs. 1: Clock supply to SSI1 halted.
1	—	1	R	Reserved This bit is always read as 1. The write value should always be 0.
0	CKDV3	1	R/W	SSI Clock Select Selects division ratio for oversample clock input to SSI 0: ×1/4 times 1: × 1 time

25.2.6 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 or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

When an RAME bit is set to 1, the corresponding on-chip RAM area is enabled. When an RAME bit is cleared to 0, the corresponding on-chip RAM area cannot be accessed. In this case, an undefined value is returned when reading data or fetching an instruction from the on-chip RAM, and writing to the on-chip RAM is ignored. The initial value of an RAME bit is 1.

Note that when clearing the RAME bit to 0 to disable the on-chip RAM, be sure to execute an instruction to read from or write to the same arbitrary address in each page before setting the RAME bit. If such an instruction is not executed, the data last written to each page may not be written to the on-chip RAM. Furthermore, an instruction to access the on-chip RAM should not be located immediately after the instruction to write to SYSCR1. If an on-chip RAM access instruction is set, normal access is not guaranteed.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RAME1	RAME0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	RAME1	1	R/W	RAM Enable 1 (corresponding RAM addresses: H'FFF84000 to H'FFF87FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled
0	RAME0	1	R/W	RAM Enable 0 (corresponding RAM addresses: H'FFF80000 to H'FFF83FFF) 0: On-chip RAM disabled 1: On-chip RAM enabled

25.2.7 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 or in deep standby mode 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 placed immediately after the instruction to write to SYSCR2. If an on-chip RAM access instruction is set, normal access is not guaranteed.

Note: When writing to this register, see section 25.4, Usage Note.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RAM WE1	RAM WE0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
1	RAMWE1	1	R/W	RAM Write Enable 1 (corresponding RAM addresses: H'FFF84000 to H'FFF87FFF) 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'FFF83FFF) 0: On-chip RAM write disabled 1: On-chip RAM write enabled

25.2.8 RAM Retaining Area Specifying Register (RAMKP)

RAMKP is an 8-bit readable/writable register that specifies whether or not to retain data in the corresponding on-chip RAM area in deep standby mode. RAMKP is initialized to H'00 by a power-on reset or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

When an RAMKP bit is set to 1, data in the corresponding on-chip RAM area is retained in deep standby mode. When an RAMWE bit is cleared to 0, data in the corresponding on-chip RAM is not retained in deep standby mode.

Deep standby mode is canceled by an interrupt (NMI or IRQ) or a reset (manual reset or power-on reset). However, when deep standby mode is canceled by a power-on reset, the contents in the corresponding on-chip RAM area are not retained even with the RAMKP bit set to 1.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMKP3	RAMKP2	RAMKP1	RAMKP0
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	RAMKP3	0	R/W	RAM Retaining Area 3 (corresponding RAM addresses: H'FFF86000 to H'FFF87FFF) 0: Data in RAM is not retained in deep standby mode 1: Data in RAM is retained in deep standby mode
2	RAMKP2	0	R/W	RAM Retaining Area 2 (corresponding RAM addresses: H'FFF84000 to H'FFF85FFF) 0: Data in RAM is not retained in deep standby mode 1: Data in RAM is retained in deep standby mode
1	RAMKP1	0	R/W	RAM Retaining Area 1 (corresponding RAM addresses: H'FFF82000 to H'FFF83FFF) 0: Data in RAM is not retained in deep standby mode 1: Data in RAM is retained in deep standby mode
0	RAMKP0	0	R/W	RAM Retaining Area 0 (corresponding RAM addresses: H'FFF80000 to H'FFF81FFF) 0: Data in RAM is not retained in deep standby mode 1: Data in RAM is retained in deep standby mode

25.2.9 Deep Standby Oscillation Settling Clock Select Register (DSCNT)

DSCNT is an 8-bit readable/writable register that selects the clock used to count the oscillation settling time when the system returns from deep standby mode. DSCNT is initialized to H'00 by a power-on reset or in deep standby mode but retains its previous value by a manual reset or in software standby mode. Only byte access is valid.

Since the frequency control register for the CPG (FRQCR) is initialized in deep standby mode, the frequency of the peripheral clock (Pϕ) specified by the CKS[2:0] bits in DSCNT is determined by the FRQCR's initial value.

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

Bit	Bit Name	Initial Value	R/W	Description
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CKS[2:0]	000	R/W	Clock Select Selects the clock used to count the oscillation settling time from among eight types clocks derived by dividing the peripheral clock (Pϕ). The oscillation settling time is calculated as follows: Oscillation settling time = 1/Pϕ × Division ratio specified by CKS[2:0] × 255 [μs] The following are the oscillation settling times when the peripheral clock (Pϕ) is running at 5, 10, and 15 MHz.

Setting value	Clock select	Oscillation settling time (ms) 5 MHz	10 MHz	15 MHz
000:	1 × Pϕ*1	0.05	0.03	0.02
001:	1/64 × Pϕ*1	3.26	1.63	1.09
010:	1/128 × Pϕ*1	6.53	3.26	2.18
011:	1/256 × Pϕ*2	13.06	6.53	4.35
100:	1/512 × Pϕ*2	26.11	13.06	8.70
101:	1/1024 × Pϕ	52.22	26.11	17.41
110:	1/4096 × Pϕ	208.90	104.45	69.63
111:	1/16384 × Pϕ	835.58	417.79	278.53

Notes: 1. Do not use this setting.
2. Set the clock so that it is equal to or longer than the oscillation settling time 2 on return from standby (t_{OSC3}).

25.2.10 Deep Standby Cancel Source Flag Register (DSFR)

DSFR is a 16-bit readable/writable register composed of two types of bits. One is the flag that confirms which interrupt canceled deep standby mode. The other is the bit that releases the retaining state of pins after canceling the deep standby mode. DSFR is initialized to H'0000 by a power-on reset by the $\overline{\text{RES}}$ pin but retains its previous value through a power-on reset caused by a WDT overflow, a manual reset, or a period in software standby mode. When deep standby mode is canceled by interrupts (NMI or IRQ) and a manual reset, this register retains the previous data although power-on reset exception handling is executed. Only word access is valid.

Since interrupt inputs for the NMI and IRQ pins specified by the interrupt controller (INTC) and the pin function controller (PFC) are always detected, these interrupts set flags even during normal operation. Therefore, all flags must be cleared immediately before the transition to deep standby mode.

If an interrupt occurred immediately before executing the SLEEP instruction after the flag clear, the system enters deep standby mode with the flag set again. To prevent this, clear the flag in DSFR even in interrupt exception handling routine.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOKEEP	—	—	—	—	—	MRESF	NMIF	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/(W)*	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written after reading 1 to clear the flag.
Even when IRQ is input after a manual reset has been accepted as a source canceling deep standby, the IRQ flag is not set.

Bit	Bit Name	Initial Value	R/W	Description
15	IOKEEP	0	R/(W)*	Pin State Retention Releases the retaining state of pins after canceling the deep standby mode 0: Pin state not retained [Clearing condition] • Writing 0 after reading 1 1: Retains pin state [Setting condition] • When transits to deep standby mode
14 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
9	MRESF	0	R/(W)*	MRES Flag 0: No interrupt on $\overline{\text{MRES}}$ pin 1: Interrupt on $\overline{\text{MRES}}$ pin
8	NMIF	0	R/(W)*	NMI Flag 0: No interrupt on NMI pin 1: Interrupt on NMI pin
7	IRQ7F	0	R/(W)*	IRQ7 Flag 0: No interrupt on IRQ7 pin 1: Interrupt on IRQ7 pin
6	IRQ6F	0	R/(W)*	IRQ6 Flag 0: No interrupt on IRQ6 pin 1: Interrupt on IRQ6 pin
5	IRQ5F	0	R/(W)*	IRQ5 Flag 0: No interrupt on IRQ5 pin 1: Interrupt on IRQ5 pin
4	IRQ4F	0	R/(W)*	IRQ4 Flag 0: No interrupt on IRQ4 pin 1: Interrupt on IRQ4 pin
3	IRQ3F	0	R/(W)*	IRQ3 Flag 0: No interrupt on IRQ3 pin 1: Interrupt on IRQ3 pin
2	IRQ2F	0	R/(W)*	IRQ2 Flag 0: No interrupt on IRQ2 pin 1: Interrupt on IRQ2 pin
1	IRQ1F	0	R/(W)*	IRQ1 Flag 0: No interrupt on IRQ1 pin 1: Interrupt on IRQ1 pin
0	IRQ0F	0	R/(W)*	IRQ0 Flag 0: No interrupt on IRQ0 pin 1: Interrupt on IRQ0 pin

Note: * Only 0 can be written after reading 1 to clear the flag.
Even when IRQ is input after a manual reset has been accepted as a source canceling deep standby, the IRQ flag is not set.

25.3 Operation

25.3.1 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 continue to be output on the CKIO pin.

(2) Canceling Sleep Mode

Sleep mode is canceled by an interrupt (NMI, H-UDI, IRQ, PINT, and on-chip peripheral module), a bus error, or a reset (manual reset or power-on reset).

- Canceling with an interrupt
  When an NMI, H-UDI, IRQ, PINT, 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 bus error
  When a bus error occurs, sleep mode is canceled and bus error exception handling is executed.
- Canceling with a reset
  Sleep mode is canceled by a power-on reset or a manual reset.

25.3.2 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 and DEEP bit in STBCR are 1 and 0 respectively. In software standby mode, not only the CPU but also the clock and on-chip peripheral modules halt. The clock output from the CKIO pin also halts in clock mode 0 or 2.

The contents of the CPU and cache registers remain unchanged. Some registers of on-chip peripheral modules are, however, initialized. As for the states of on-chip peripheral module registers in software standby mode, see section 28.3, Register States in Each Operating 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.

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 and DEEP bits in STBCR to 1 and 0 respectively, read STBCR. Then, execute a SLEEP instruction.

(2) Canceling Software Standby Mode

Software standby mode is canceled by interrupts (NMI or IRQ) or a reset (manual reset or power-on reset). In clock modes 0 and 2, a clock signal starts to be output from the CKIO pin.

- Canceling with an interrupt

  When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) of the interrupt controller (INTC)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0) (selected by the IRQn sense select bits (IRQn1S and IRQn0S) in interrupt control register 1 (ICR1) of the interrupt controller (INTC)) is detected, clock oscillation is started. This clock 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 thus cleared and NMI interrupt exception handling (IRQ interrupt exception handling in case of IRQ) is started. However, when the IRQ interrupt priority level is lower than the interrupt mask level set in the status register (SR) of the CPU, the interrupt request is not accepted and software standby mode is not canceled.

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 CKIO 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.)

- Canceling with a reset

  When the $\overline{\text{RES}}$ pin is driven low, software standby mode is canceled and the LSI enters the power-on reset state. After that, if the $\overline{\text{RES}}$ pin is driven high, the power-on reset exception handling is started.

  When the $\overline{\text{MRES}}$ pin is driven low, software standby mode is canceled and the LSI enters the manual reset state. After that, if the $\overline{\text{MRES}}$ pin is driven high, the manual reset exception handling is started.

  Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation settles. The internal clock will continue to be output to the CKIO pin in clock mode 0 or 2.

(3) Note on Making a Transition To Software Standby Mode

If the SLEEP instruction is executed to make a transition to software standby mode during transfer by the DMAC, the DMAC stops its operation without waiting for the completion of the transfer. Thus, the DMA transfer is not guaranteed. Therefore, when making a transition to software standby mode, wait for the completion of the DMA transfer or stop the DMA transfer to execute the SLEEP instruction.

25.3.3 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 25.1.

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 and DEEP bits in STBCR are set to 1 and 0 respectively, 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.

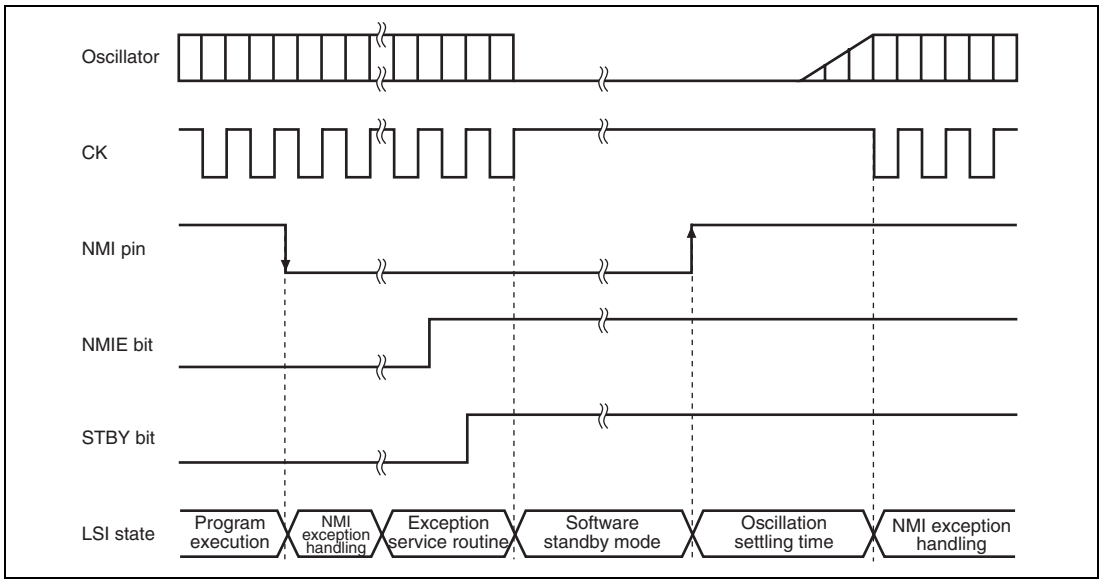


Figure 25.1 NMI Timing in Software Standby Mode (Application Example)

25.3.4 Deep Standby Mode

(1) Transition to Deep Standby Mode

The LSI switches from a program execution state to deep standby mode by executing the SLEEP instruction when the STBY bit and DEEP bit in STBCR are set to 1. In deep standby mode, not only the CPU, clocks, and on-chip peripheral modules but also power supply is turned off excluding the on-chip RAM retaining area specified by the RAMKP3 to RAMKP0 bits in the RAMKP register and RTC. This can significantly reduce power consumption. Therefore, data in the registers of the CPU, cache, and on-chip peripheral modules are not retained. Pin state values immediately before the transition to deep standby mode can be retained.

The CPU takes one cycle to finish writing to DSFR, and then executes processing for the next instruction. However, it actually takes one or more cycles to write. Therefore, execute a SLEEP instruction after reading DSFR to have the values written to DSFR by the CPU to be definitely reflected in the SLEEP instruction.

The procedure for switching to deep standby mode is as follows. Figure 25.2 also shows its flowchart.

1. Set the RAMKP3 to RAMKP0 bits in the RAMKP register for the corresponding on-chip RAM retaining area.
2. Execute read and write of an arbitrary but the same address for each page in the retaining RAM area. When this is not executed, data last written may not be stored in the on-chip RAM. If there is a write to the on-chip RAM after this time, execute this processing after the last write to the on-chip RAM.
3. Set the CKS[2:0] bits in the DSCNT register so that the initial value of FRQCR in the CPG becomes larger than the oscillation settling time.
4. Set the STBY and DEEP bits in the STBCR register to 1.
5. Read out the DSFR register after clearing the flag in the DSFR register. Then execute the SLEEP instruction.

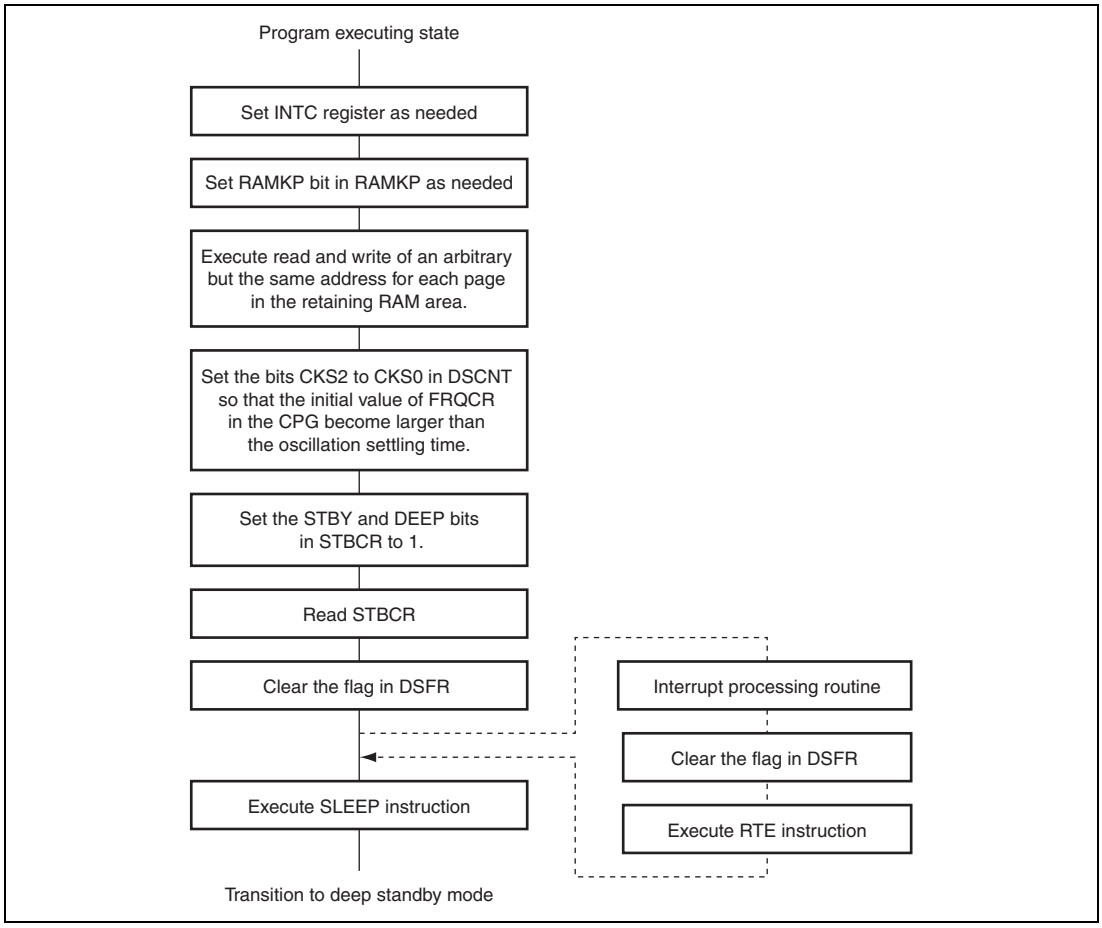


Figure 25.2 Flowchart of Transition to Deep Standby Mode

(2) Canceling Deep Standby Mode

Deep standby mode is canceled by interrupts (NMI or IRQ) or a reset (manual reset or power-on reset). However, IRQ is canceled only by PE7 to PE4 and PC25 to PC22. To cancel deep standby mode by interrupt NMI or IRQ, a power-on reset exception handling instead of an interrupt exception handling is executed. In the same way, a power-on reset exception handling is executed by a power-on reset. Figure 25.3 shows the flowchart of canceling deep standby mode.

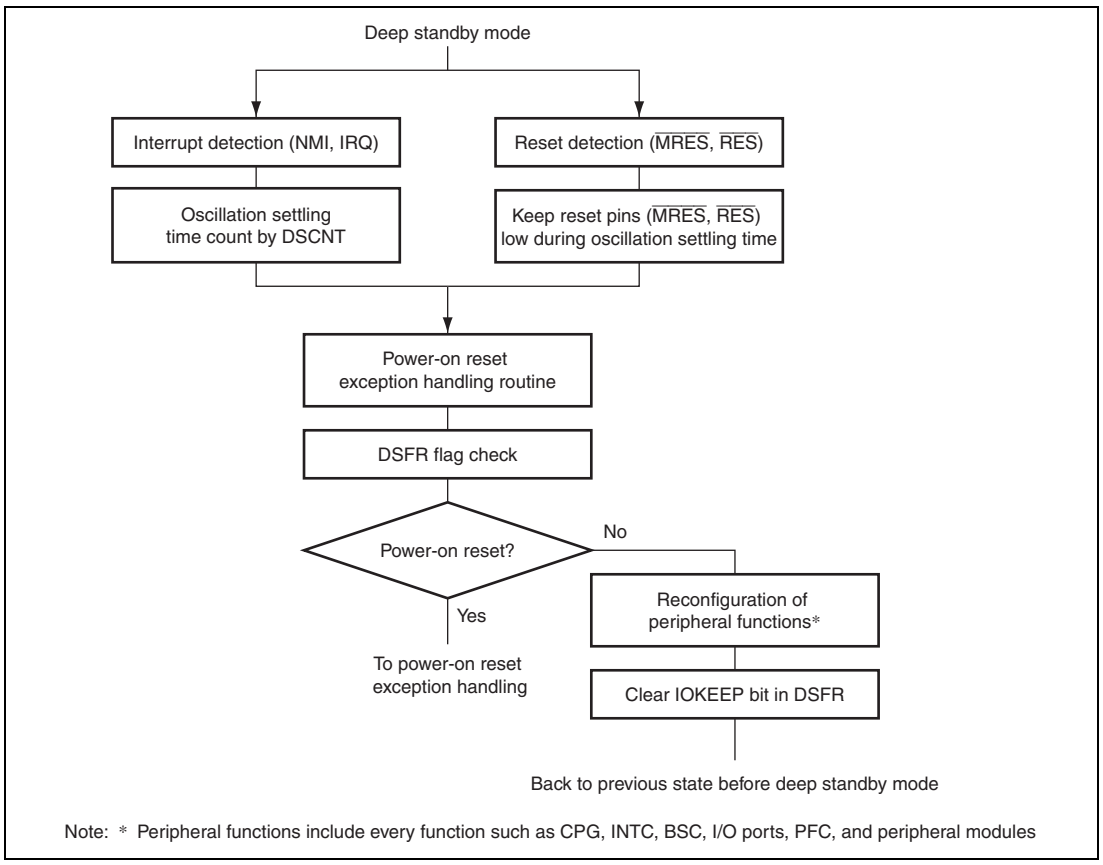


Figure 25.3 Flowchart of Canceling Deep Standby Mode

- Canceling with an interrupt

  When the falling edge or rising edge of the NMI pin (selected by the NMI edge select bit (NMIE) in interrupt control register 0 (ICR0) of the interrupt controller (INTC)) or the falling edge or rising edge of an IRQ pin (IRQ7 to IRQ0: PE7 to PE4 and PC25 to PC22) (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 after the wait time for the oscillation settling time. This clock pulse is supplied only to the oscillation settling counter (DSCNT) used to count the oscillation settling time.

  After the elapse of the time set in the clock select bits (CKS[2:0]) in DSCNT before the transition to deep standby mode, an 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. Deep standby mode is thus cleared and reset exception handling is started.

  When canceling deep standby mode by the NMI interrupt or IRQ interrupt, set the CKS[2:0] bits so that the overflow period will be equal to or longer than the oscillation settling time.

  The clock output phase of the CKIO pin may be unstable immediately after detecting an interrupt and until deep standby mode is canceled. When deep standby mode is canceled by the falling edge of the NMI pin, the NMI pin should be high when the CPU enters deep standby mode (when the clock pulse stops) and should be low when the CPU returns from deep standby mode (when the clock is initiated after the oscillation settling). When deep standby mode is canceled by the rising edge of the NMI pin, the NMI pin should be low when the CPU enters deep standby mode (when the clock pulse stops) and should be high when the CPU returns from deep standby mode (when the clock is initiated after the oscillation settling). (This is the same with the IRQ pin.)

- Canceling with a reset

  When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin is driven low, this LSI enters the power-on reset state and deep standby mode is canceled.

  Keep the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low until the clock oscillation settles. When deep standby mode is canceled by the $\overline{\text{RES}}$ pin, the contents in the on-chip RAM area are not retained.

(3) Operation after Canceling Deep Standby Mode

When deep standby mode is canceled by interrupts (NMI or IRQ) or a manual reset, the deep standby cancel source flag register (DSFR) can be used to confirm which interrupt has canceled the mode.

Pins retain the state immediately before the transition to deep standby mode. However, in canceling deep standby mode, only the pins in buses listed in the table 25.3 can fetch programs while canceling pin states. Pins other than those retain the pin states after canceling deep standby mode, in which DSFR can confirm which interrupt has triggered returning to deep standby mode. Reconfiguration of peripheral functions is required to return to the previous state of deep standby mode. Peripheral functions include every function such as CPG, INTC, BSC, I/O ports, PFC, and peripheral modules. After the reconfiguration, pin-retaining state can be canceled by reading 1 in the IOKEEP bit of DSFR then writing 0 to it.

Table 25.3 Pin States in Different Modes

Operation Mode (1) (External 8_bit Bus Initiated)	Operation Mode (2) (External 16_bit Bus Initiated)	Operation Mode (3) (External 32_bit Bus Initiated)
PA[23:0]	PA[23:0]	PA[23:0]
PB[7:0]	PB[15:0]	PB[31:0]
PC[9:8], PC[0]	PC[10:8], PC[0]	PC[12:8], PC[0]
CKIO	CKIO	CKIO

(4) Note on Making a Transition To Deep Standby Mode

If the SLEEP instruction is executed to make a transition to deep standby mode during transfer by the DMAC, the DMAC stops its operation without waiting for the completion of the transfer. Thus, the DMA transfer is not guaranteed. Therefore, when making a transition to deep standby mode, wait for the completion of the DMA transfer or stop the DMA transfer to execute the SLEEP instruction.

25.3.5 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 the 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, see table 25.4.

However, the state of DAC registers are exceptional. 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 RTC, H-UDI, UBC, DMAC, and AUD-II). 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.

25.4 Usage Note

25.4.1 Note on Setting Registers

When writing to the registers related to power-down modes, note the following.

When writing to the register related to power-down modes, the CPU, after executing a write instruction, executes the next instruction without waiting for the write operation to complete. Therefore, to reflect the change specified by writing to the register while the next instruction is executed, insert a dummy read of the same register between the register write instruction and the next instruction.

25.4.2 Note on Canceling Standby Mode when an External Clock is being Input

When release from standby mode is initiated by an interrupt (NMI or IRQ) while an external clock from the EXTAL pin or CKIO pin is in use, make sure that the external clock is being input before input of the interrupt. If this is not the case, correct counting of the oscillation settling time will not be possible.

Section 26 User Debugging Interface (H-UDI)

This LSI incorporates a user debugging interface (H-UDI) for emulator support.

26.1 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 26.1 shows a block diagram of the H-UDI.

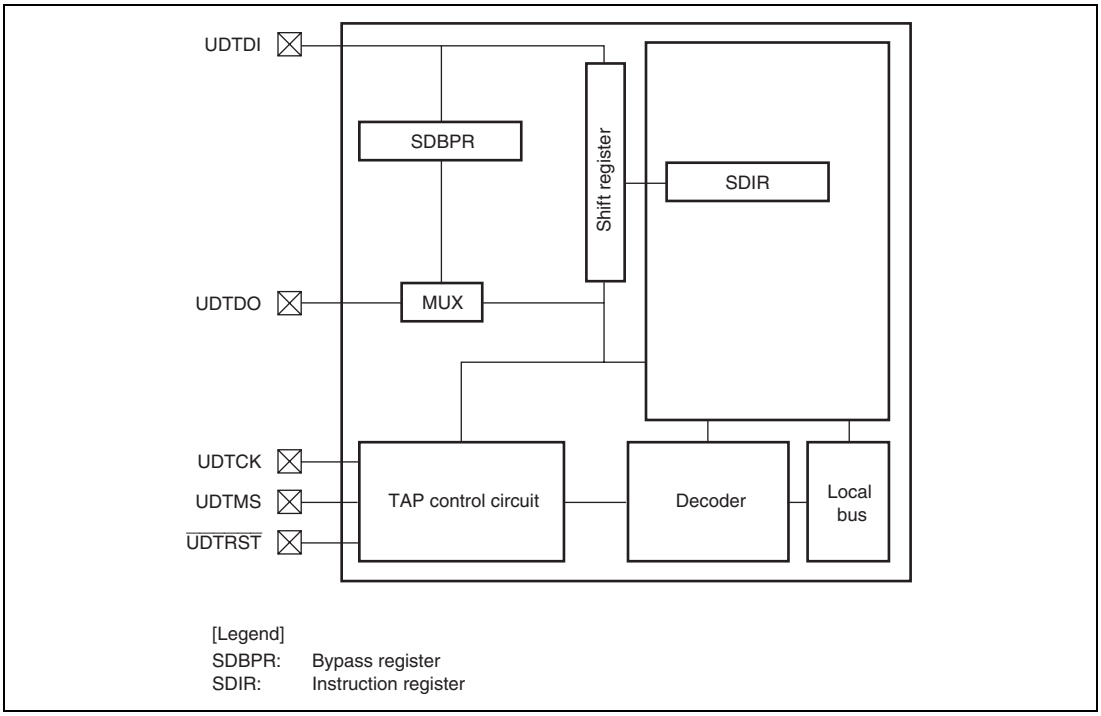


Figure 26.1 Block Diagram of H-UDI

26.2 Input/Output Pins

Table 26.1 Pin Configuration

Pin Name	Symbol	I/O	Function
H-UDI serial data input/output clock pin	UDTCK*	Input	Data is serially supplied to the H-UDI from the data input pin (UDTDI), and output from the data output pin (UDTDO), in synchronization with this clock. Fix high when not used.
Mode select input pin	UDTMS*	Input	The state of the TAP control circuit is determined by changing this signal in synchronization with UDTCK. For the protocol, see figure 26.2. Fix high when not used.
H-UDI reset input pin	$\overline{\text{UDTRST}}$*	Input	Input is accepted asynchronously with respect to UDTCK, and when low, the H-UDI is reset. $\overline{\text{UDTRST}}$ must be low for oscillation settling time when power is turned on. See section 26.4.2, Reset Types, for more information.
H-UDI serial data input pin	UDTDI*	Input	Data transfer to the H-UDI is executed by changing this signal in synchronization with UDTCK. Fix high when not used.
H-UDI serial data output pin	UDTDO	Output	Data read from the H-UDI is executed by reading this pin in synchronization with UDTCK. The initial value of the data output timing is the UDTCK falling edge. This can be changed to the UDTCK rising edge by inputting the UDTDO change timing switch command to SDIR. See section 26.4.3, UDTDO Output Timing, for more information.
ASE mode select pin	$\overline{\text{ASEMD}}$	Input	Fix high.

Note: * The pin with the pull-up function.

26.3 Register Descriptions

The H-UDI has the following registers.

Table 26.2 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
Bypass register	SDBPR	—	—	—	—
Instruction register	SDIR	R	H'EFFD	H'FFFD9000	16

26.3.1 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 UDTDI and UDTDO. The initial value is undefined.

26.3.2 Instruction Register (SDIR)

SDIR is a 16-bit read-only register. It is initialized by $\overline{\text{UDTRST}}$ assertion, in the TAP test-logic-reset state or in deep standby mode, and can be written to by the H-UDI irrespective of the 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 UDTDI. For commands, see table 26.3.
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 26.3 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	UDTDO change timing switch
1	0	1	1	—	—	—	—	H-UDI interrupt
1	1	1	1	—	—	—	—	BYPASS mode
Other than above								Reserved

26.4 Operation

26.4.1 TAP Controller

Figure 26.2 shows the internal states of the TAP controller.

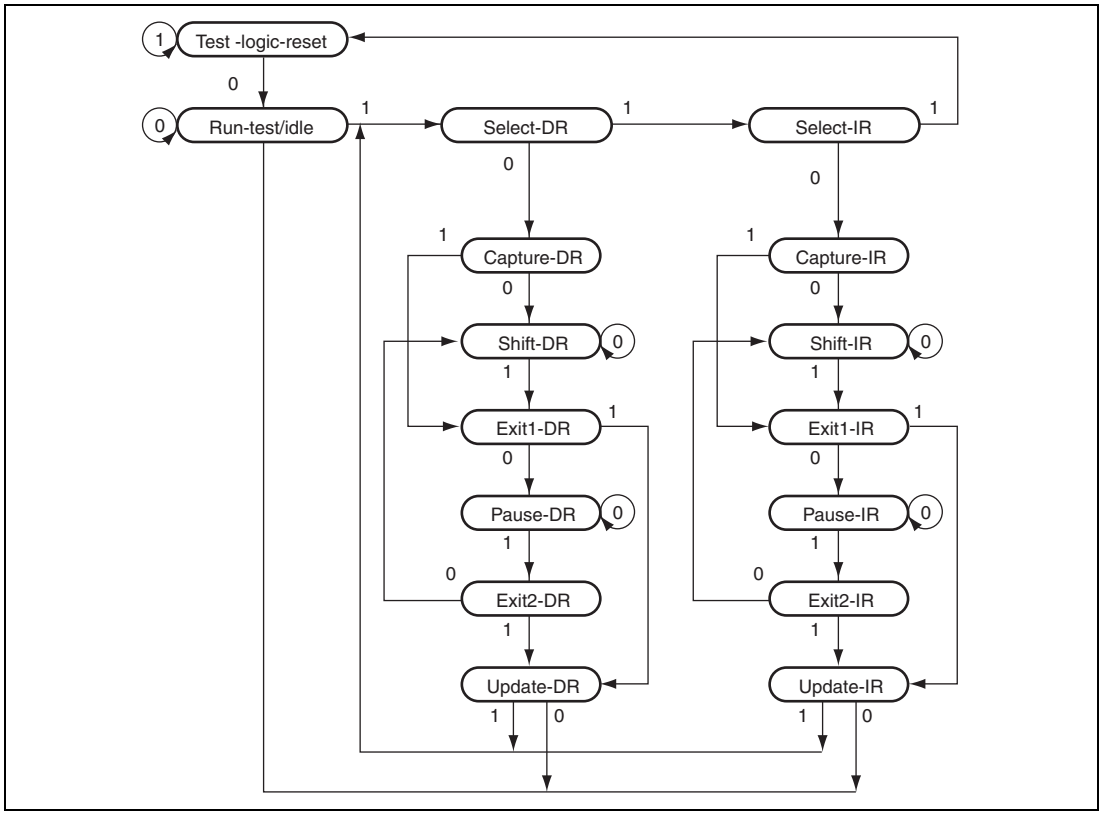


Figure 26.2 TAP Controller State Transitions

Note: The transition condition is the UDTMS value at the rising edge of UDTCK. The UDTDI value is sampled at the rising edge of UDTCK; shifting occurs at the falling edge of UDTCK. For details on change timing of the UDTDO value, see section 26.4.3, UDTDO Output Timing. The UDTDO is at high impedance, except with shift-DR and shift-IR states. There is a transition to test-logic-reset asynchronously with UDTCK by $\overline{\text{UDTRST}}$ assertion or deep standby mode.

26.4.2 Reset Types

Table 26.4 Reset Types

$\overline{\text{ASEMD}}$*	$\overline{\text{RES}}$	$\overline{\text{UDTRST}}$	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

Note: * Fix ASEMD to high.

26.4.3 UDTDO Output Timing

The initial value of the UDTDO change timing is to perform data output from the UDTDO pin on the UDTCK falling edge. However, setting a UDTDO change timing switch command in SDIR via the H-UDI pin and passing the Update-IR state synchronizes the UDTDO change timing to the UDTCK rising edge. Hereafter, to synchronize the UDTDO change timing with the UDTCK falling edge, the $\overline{\text{UDTRST}}$ pin must be asserted simultaneously with a power-on reset or deep standby mode must be entered. In the case of a power-on reset by the $\overline{\text{RES}}$ pin, the LSI falls in reset state for a certain period after the $\overline{\text{RES}}$ pin negation. Therefore, when the $\overline{\text{UDTRST}}$ pin is asserted immediately after the $\overline{\text{RES}}$ pin negation, a UDTDO change timing switch command is cleared and the UDTDO change timing becomes synchronous with the output of UDTCK falling edge. To prevent this, at least 20 tcyc must be set between the change timings of the $\overline{\text{RES}}$ pin and $\overline{\text{UDTRST}}$ pin.

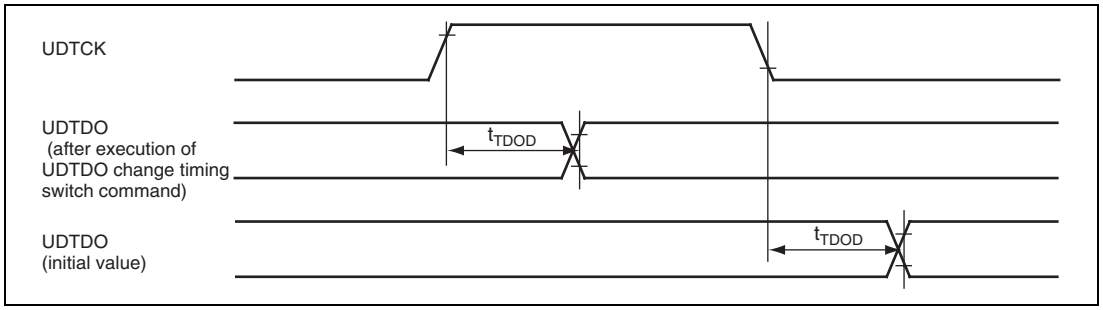


Figure 26.3 H-UDI Data Transfer Timing

26.4.4 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.

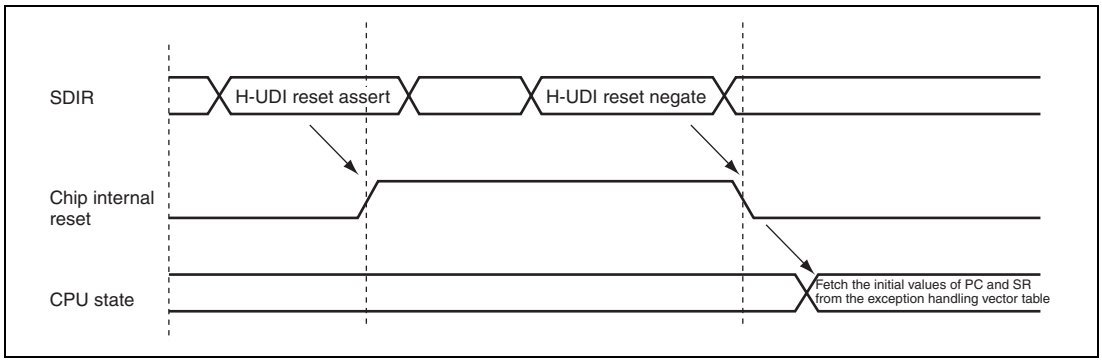


Figure 26.4 H-UDI Reset

26.4.5 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 or deep standby mode.

26.5 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 or H-UDI module standby state, all of the functions in the H-UDI cannot be used. To retain the TAP status before and after software standby mode or H-UDI module standby state, keep UDTCK high before entering software standby mode or H-UDI module standby state.
3. In deep standby mode, all of the functions in the H-UDI cannot be used. H-UDI is initialized in deep standby mode.
4. If the $\overline{\text{UDTRST}}$ pin is asserted immediately after the setting of the UDTDO transition timing switching command and the negation of the RES pin, the UDTDO transition timing switching command is cleared. To avoid this case, make sure to put 20 tcyc or longer between the signal transition timing of the $\overline{\text{RES}}$ and $\overline{\text{UDTRST}}$ pins. For details, see section 26.4.3, UDTDO Output Timing.
5. When starting the TAP controller after the negation of the $\overline{\text{UDTRST}}$ pin, make sure to allow 200 ms or more after the negation.

Section 27 Advanced User Debugger II (AUD-II)

The AUD-II offers functions that support user program debugging with the LSI mounted and operated in actual performance. Use of the AUD-II simplifies the construction of a simple emulator, with functions such as monitoring/tuning of on-chip RAM data.

27.1 Features

The AUD-II can be used in RAM monitor mode by setting AUDMD.

RAM monitor mode:

- Functions to read/write modules connected to internal/external buses (except cache and H-UDI)
- Outputs data corresponding to an address that is externally written to AUDATA
- Transmits data to the address in AUDATA to which address and data are written

27.2 Input/Output Pins

Table 27.1 Pin Configuration

Pin Name	Symbol	Function
AUD reset	$\overline{\text{AUDRST}}$	AUD reset input
AUD sync signal	$\overline{\text{AUDSYNC}}$	Data start position identification signal input
AUD clock	AUDCK	External clock input
AUD mode	AUDMD	Mode select input (H)
AUD data	AUDATA[3:0]	Monitor address input and data input/output

(1) Description of Pins

Table 27.2 Description of Pins

Pin	Function
AUDMD	The mode is selected by changing the input level at this pin. Low: Setting prohibited High: RAM monitor mode The input at this pin should be changed when $\overline{\text{AUDRST}}$ is low.
$\overline{\text{AUDRST}}$	When this pin is driven low, the AUD enters the reset state and the AUD's internal buffers and logic are reset. When $\overline{\text{AUDRST}}$ goes high again after the AUDMD level settles, the AUD starts operating in the selected mode.
AUDCK	This pin is for external clock input. Input the clock to be used for debugging. Note that the available frequency is up to Bϕ/2.
$\overline{\text{AUDSYNC}}$	AUD Bus Command Valid Signal 1: Read data is output 0: Inputs write address, data, DIR command Note: Do not assert this pin until commands are input to AUDATA from outside and necessary data is prepared. For details, see the protocol as described later.
AUDATA[3:0]	The following data is output in time-sharing mode. • AUD bus command • Address • Data When a command is input from outside, data is output after Ready is transmitted. The output starts after $\overline{\text{AUDSYNC}}$ is negated. For details, see the protocol as described later.

27.3 RAM Monitor Mode

In this mode, all the modules connected to this LSI's internal or external bus can be read and written to (except cache and H-UDI), allowing RAM monitoring and tuning to be carried out.

27.3.1 Communication Protocol

The AUD-II latches the AUDATA input when $\overline{\text{AUDSYNC}}$ is asserted. The following AUDATA input format should be used.

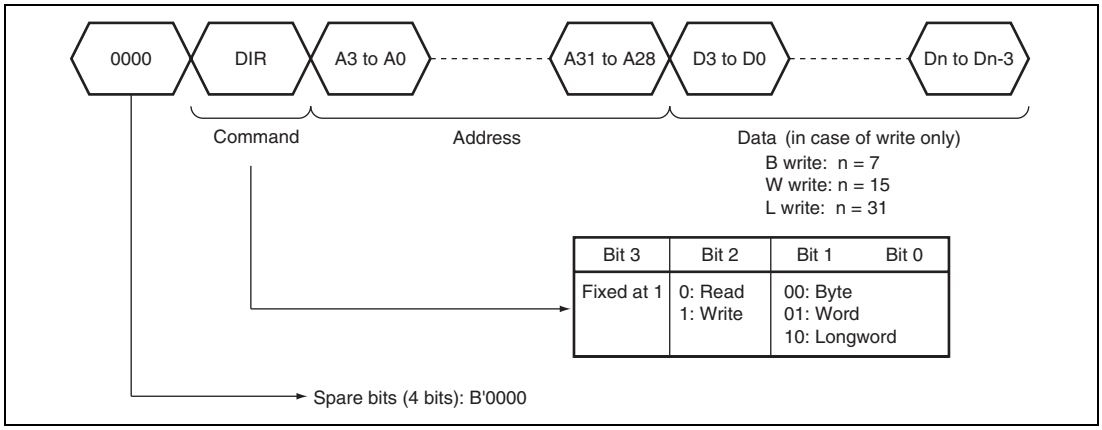


Figure 27.1 AUDATA Input Format

27.3.2 Operation

Operation starts in RAM monitor mode when $\overline{\text{AUDRST}}$ is asserted, AUDMD is driven high, and then $\overline{\text{AUDRST}}$ is negated.

Figure 27.2 shows an example of a read operation, and figure 27.3 shows an example of a write operation.

When $\overline{\text{AUDSYNC}}$ is asserted, input from the AUDATA pins begins. When a command, address, or data (writing only) is input in the format shown in figure 27.1, execution of read/write access to the specified address is started. During internal execution, the AUD returns Not Ready (B'0000). When execution is completed, the Ready flag (B'0001) is returned (figures 27. 2 and 27. 3). Table 27.3 shows the Ready flag format.

In a read, data of the specified size is output when $\overline{\text{AUDSYNC}}$ is negated following detection of this flag (figure 27. 2).

If a command other than the above is input in DIR, the AUD-II treats this as a command error, disables processing, and sets bit 1 in the Ready flag to 1. If a read/write operation initiated by the command specified in DIR causes a bus error, the AUD-II disables processing and sets bit 2 in the Ready flag to 1 (figure 27. 4).

Bus error conditions are shown below.

1. Word access to address 4n+1 or 4n+3
2. Longword access to address 4n+1, 4n+2, or 4n+3

Table 27.3 Ready Flag Format

Bit 3	Bit 2	Bit 1	Bit 0
Fixed at 0	0: Normal status 1: Bus error	0: Normal status 1: Command error	0: Not ready 1: Ready

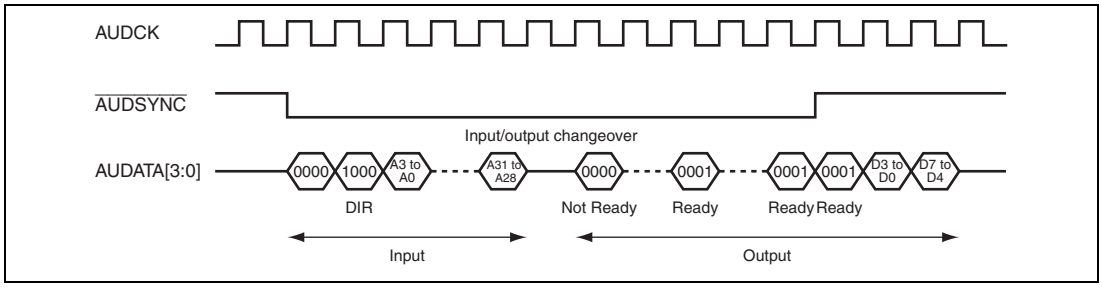


Figure 27.2 Example of Read Operation (Byte Read)

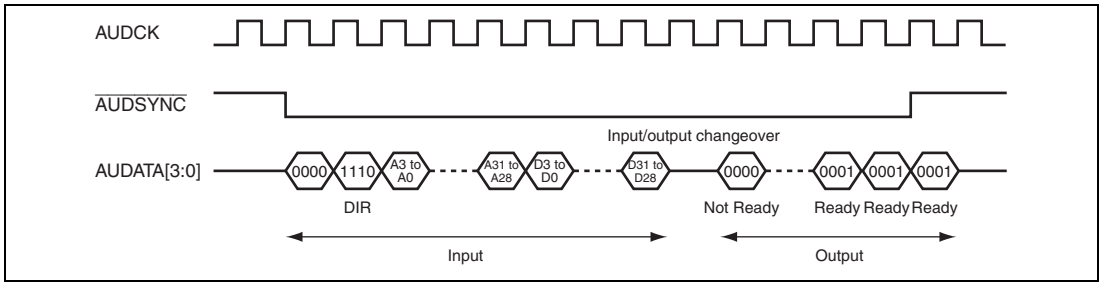


Figure 27.3 Example of Write Operation (Longword Write)

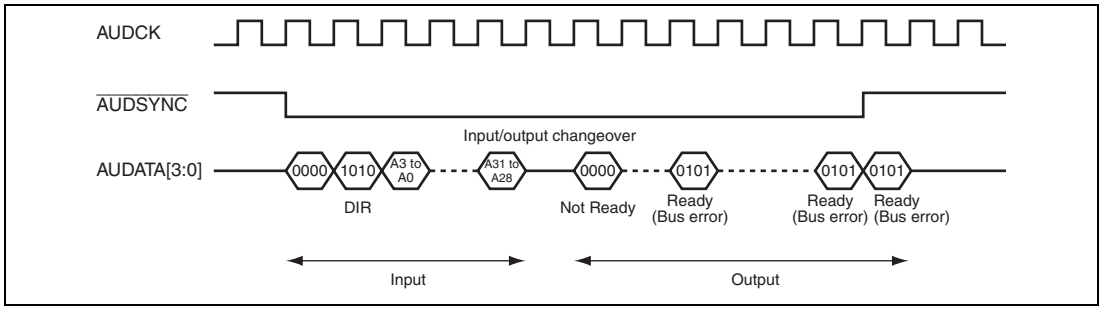


Figure 27.4 Example of Error Occurrence (Longword Read)

27.3.3 Usage Notes (RAM Monitor Mode)

(1) Guidelines for initialization of the RAM monitor mode

The buffers in this debugger and the processing status are initialized under the following conditions.

- Power-on reset
- When the $\overline{\text{AUDRST}}$ pin is driven low
- Module standby
- Deep standby mode

(2) Guidelines for AUDCK

- AUDCK is for inputting the external clock. Input the clock to satisfy Bϕ/2 ≥ AUDCK.

(3) Other Limitations

- Do not negate $\overline{\text{AUDSYNC}}$ until the command is input to AUDATA and the Ready is returned.
- The RAM monitor functions in sleep mode but is not available in software standby or deep standby mode.

Section 28 List of Registers

The address map gives information on the on-chip I/O registers and is configured as described below.

1. Register Addresses (address order)

- Registers are described by functional module, in order of the corresponding section numbers.
- Access to reserved addresses that are not described in this register address list is prohibited.
- When addresses 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 (address order).
- 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.
- When registers consist of 16 or 32 bits, the bits are given from the MSB side. The listing order of bytes is based on big-endian mode.

3. Register States in Each Operating Mode

- Register states are described in the same order as the Register Addresses (address order).
- 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.

28.1 Register Addresses (Address Order)

Entries under Access Size indicate numbers of bits.

Note: Access to undefined or reserved addresses is prohibited. Since operation or continued operation is not guaranteed when these registers are accessed, do not attempt such access.

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Bus monitor enable register	SYCBEEN	32	H'FF400000	Bus Monitor	8, 16, 32
Bus monitor status register 1	SYCBESTS1	32	H'FF400004		8, 16, 32
Bus monitor status register 2	SYCBESTS2	32	H'FF400008		8, 16, 32
Bus error control register	SYCBESW	32	H'FF40000C		8, 16, 32
CS0 control register	CS0CNT	32	H'FF420000	BSC	8, 16, 32
CS0 recovery cycle setting register	CS0REC	32	H'FF420008		8, 16, 32
CS1 control register	CS1CNT	32	H'FF420010		8, 16, 32
CS1 recovery cycle setting register	CS1REC	32	H'FF420018		8, 16, 32
CS2 control register	CS2CNT	32	H'FF420020		8, 16, 32
CS2 recovery cycle setting register	CS2REC	32	H'FF420028		8, 16, 32
CS3 control register	CS3CNT	32	H'FF420030		8, 16, 32
CS3 recovery cycle setting register	CS3REC	32	H'FF420038		8, 16, 32
CS4 control register	CS4CNT	32	H'FF420040		8, 16, 32
CS4 recovery cycle setting register	CS4REC	32	H'FF420048		8, 16, 32
CS5 control register	CS5CNT	32	H'FF420050		8, 16, 32
CS5 recovery cycle setting register	CS5REC	32	H'FF420058		8, 16, 32
CS6 control register	CS6CNT	32	H'FF420060		8, 16, 32
CS6 recovery cycle setting register	CS6REC	32	H'FF420068		8, 16, 32
SDRAMC0 control register	SDC0CNT	32	H'FF420100		8, 16, 32
SDRAMC1 control register	SDC1CNT	32	H'FF420110		8, 16, 32
CS0 mode register	CSMOD0	32	H'FF421000		8, 16, 32
CS0 wait control register 1	CS1WCNT0	32	H'FF421004		8, 16, 32
CS0 wait control register 2	CS2WCNT0	32	H'FF421008		8, 16, 32
CS1 mode register	CSMOD1	32	H'FF421010		8, 16, 32
CS1 wait control register 1	CS1WCNT1	32	H'FF421014		8, 16, 32
CS1 wait control register 2	CS2WCNT1	32	H'FF421018		8, 16, 32
CS2 mode register	CSMOD2	32	H'FF421020		8, 16, 32
CS2 wait control register 1	CS1WCNT2	32	H'FF421024		8, 16, 32
CS2 wait control register 2	CS2WCNT2	32	H'FF421028		8, 16, 32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
CS3 mode register	CSMOD3	32	H'FF421030	BSC	8, 16, 32
CS3 wait control register 1	CS1WCNT3	32	H'FF421034		8, 16, 32
CS3 wait control register 2	CS2WCNT3	32	H'FF421038		8, 16, 32
CS4 mode register	CSMOD4	32	H'FF421040		8, 16, 32
CS4 wait control register 1	CS1WCNT4	32	H'FF421044		8, 16, 32
CS4 wait control register 2	CS2WCNT4	32	H'FF421048		8, 16, 32
CS5 mode register	CSMOD5	32	H'FF421050		8, 16, 32
CS5 wait control register 1	CS1WCNT5	32	H'FF421054		8, 16, 32
CS5 wait control register 2	CS2WCNT5	32	H'FF421058		8, 16, 32
CS6 mode register	CSMOD6	32	H'FF421060		8, 16, 32
CS6 wait control register 1	CS1WCNT6	32	H'FF421064		8, 16, 32
CS6 wait control register 2	CS2WCNT6	32	H'FF421068		8, 16, 32
SDRAM refresh control register 0	SDRFCNT0	32	H'FF422000		8, 16, 32
SDRAM refresh control register 1	SDRFCNT1	32	H'FF422004		16, 32
SDRAM initialized register 0	SDIR0	32	H'FF422008		8, 16, 32
SDRAM initialized register 1	SDIR1	32	H'FF42200C		8, 16, 32
SDRAM power down control register	SDPWDCNT	32	H'FF422010		8, 16, 32
SDRAM deep power down control register	SDDPWDCNT	32	H'FF422014		8, 16, 32
SDRAM0 address register	SD0ADR	32	H'FF422020		8, 16, 32
SDRAM0 timing register	SD0TR	32	H'FF422024		8, 16, 32
SDRAM0 mode register	SD0MOD	32	H'FF422028		16, 32
SDRAM1 address register	SD1ADR	32	H'FF422040		8, 16, 32
SDRAM1 timing register	SD1TR	32	H'FF422044		8, 16, 32
SDRAM1 mode register	SD1MOD	32	H'FF422048		16, 32
SDRAM status register	SDSTR	32	H'FF4220E4		8, 16, 32
SDRAM clock stop control signal setting register	SDCKSCNT	32	H'FF4220E8		8, 16, 32
DMA current source address register 0	DMCSADR0	32	H'FF460000	DMAC	32
DMA current destination address register 0	DMCDADR0	32	H'FF460004		32
DMA current byte count register 0	DMCBCT0	32	H'FF460008		32
DMA mode register 0	DMMOD0	32	H'FF46000C		32
DMA current source address register 1	DMCSADR1	32	H'FF460010		32
DMA current destination address register 1	DMCDADR1	32	H'FF460014		32
DMA current byte count register 1	DMCBCT1	32	H'FF460018		32
DMA mode register 1	DMMOD1	32	H'FF46001C		32
DMA current source address register 2	DMCSADR2	32	H'FF460020		32
DMA current destination address register 2	DMCDADR2	32	H'FF460024		32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
DMA current byte count register 2	DMCBCT2	32	H'FF460028	DMAC	32
DMA mode register 2	DMMOD2	32	H'FF46002C		32
DMA current source address register 3	DMCSADR3	32	H'FF460030		32
DMA current destination address register 3	DMCDADR3	32	H'FF460034		32
DMA current byte count register 3	DMCBCT3	32	H'FF460038		32
DMA mode register 3	DMMOD3	32	H'FF46003C		32
DMA current source address register 4	DMCSADR4	32	H'FF460040		32
DMA current destination address register 4	DMCDADR4	32	H'FF460044		32
DMA current byte count register 4	DMCBCT4	32	H'FF460048		32
DMA mode register 4	DMMOD4	32	H'FF46004C		32
DMA current source address register 5	DMCSADR5	32	H'FF460050		32
DMA current destination address register 5	DMCDADR5	32	H'FF460054		32
DMA current byte count register 5	DMCBCT5	32	H'FF460058		32
DMA mode register 5	DMMOD5	32	H'FF46005C		32
DMA current source address register 6	DMCSADR6	32	H'FF460060		32
DMA current destination address register 6	DMCDADR6	32	H'FF460064		32
DMA current byte count register 6	DMCBCT6	32	H'FF460068		32
DMA mode register 6	DMMOD6	32	H'FF46006C		32
DMA current source address register 7	DMCSADR7	32	H'FF460070		32
DMA current destination address register 7	DMCDADR7	32	H'FF460074		32
DMA current byte count register 7	DMCBCT7	32	H'FF460078		32
DMA mode register 7	DMMOD7	32	H'FF46007C		32
DMA reload source address register 0	DMRSADR0	32	H'FF460200		32
DMA reload destination address register 0	DMRDADR0	32	H'FF460204		32
DMA reload byte count register 0	DMRBCT0	32	H'FF460208		32
DMA reload source address register 1	DMRSADR1	32	H'FF460210		32
DMA reload destination address register 1	DMRDADR1	32	H'FF460214		32
DMA reload byte count register 1	DMRBCT1	32	H'FF460218		32
DMA reload source address register 2	DMRSADR2	32	H'FF460220		32
DMA reload destination address register 2	DMRDADR2	32	H'FF460224		32
DMA reload byte count register 2	DMRBCT2	32	H'FF460228		32
DMA reload source address register 3	DMRSADR3	32	H'FF460230		32
DMA reload destination address register 3	DMRDADR3	32	H'FF460234		32
DMA reload byte count register 3	DMRBCT3	32	H'FF460238		32
DMA reload source address register 4	DMRSADR4	32	H'FF460240		32
DMA reload destination address register 4	DMRDADR4	32	H'FF460244		32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
DMA reload byte count register 4	DMRBCT4	32	H'FF460248	DMAC	32
DMA reload source address register 5	DMRSADR5	32	H'FF460250		32
DMA reload destination address register 5	DMRDADR5	32	H'FF460254		32
DMA reload byte count register 5	DMRBCT5	32	H'FF460258		32
DMA reload source address register 6	DMRSADR6	32	H'FF460260		32
DMA reload destination address register 6	DMRDADR6	32	H'FF460264		32
DMA reload byte count register 6	DMRBCT6	32	H'FF460268		32
DMA reload source address register 7	DMRSADR7	32	H'FF460270		32
DMA reload destination address register 7	DMRDADR7	32	H'FF460274		32
DMA reload byte count register 7	DMRBCT7	32	H'FF460278		32
DMA control register A0	DMCNTA0	32	H'FF460400		8, 16, 32
DMA control register B0	DMCNTB0	32	H'FF460404		8, 16, 32
DMA control register A1	DMCNTA1	32	H'FF460408		8, 16, 32
DMA control register B1	DMCNTB1	32	H'FF46040C		8, 16, 32
DMA control register A2	DMCNTA2	32	H'FF460410		8, 16, 32
DMA control register B2	DMCNTB2	32	H'FF460414		8, 16, 32
DMA control register A3	DMCNTA3	32	H'FF460418		8, 16, 32
DMA control register B3	DMCNTB3	32	H'FF46041C		8, 16, 32
DMA control register A4	DMCNTA4	32	H'FF460420		8, 16, 32
DMA control register B4	DMCNTB4	32	H'FF460424		8, 16, 32
DMA control register A5	DMCNTA5	32	H'FF460428		8, 16, 32
DMA control register B5	DMCNTB5	32	H'FF46042C		8, 16, 32
DMA control register A6	DMCNTA6	32	H'FF460430		8, 16, 32
DMA control register B6	DMCNTB6	32	H'FF460434		8, 16, 32
DMA control register A7	DMCNTA7	32	H'FF460438		8, 16, 32
DMA control register B7	DMCNTB7	32	H'FF46043C		8, 16, 32
DMA activation control register	DMSCNT	32	H'FF460500		8, 16, 32
DMA interrupt control register	DMICNT	32	H'FF460508		8, 16, 32
DMA common interrupt control register	DMICNTA	32	H'FF46050C		8, 16, 32
DMA interrupt status register	DMISTS	32	H'FF460510		8, 16, 32
DMA transfer end detection register	DMEDET	32	H'FF460514		8, 16, 32
DMA arbitration status register	DMASTS	32	H'FF460518		8, 16, 32
Break address register_0	BAR_0	32	H'FFFC0400	UBC	32
Break address mask register_0	BAMR_0	32	H'FFFC0404		32
Break data register_0	BDR_0	32	H'FFFC0408		32
Break data mask register_0	BDMR_0	32	H'FFFC040C		32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Break address register_1	BAR_1	32	H'FFFC0410	UBC	32
Break address mask register_1	BAMR_1	32	H'FFFC0414		32
Break data register_1	BDR_1	32	H'FFFC0418		32
Break data mask register_1	BDMR_1	32	H'FFFC041C		32
Break bus cycle register_0	BBR_0	16	H'FFFC04A0		16
Break bus cycle register_1	BBR_1	16	H'FFFC04B0		16
Break control register	BRCR	32	H'FFFC04C0		32
Cache control register 1	CCR1	32	H'FFFC1000	Cache	32
Cache control register 2	CCR2	32	H'FFFC1004		32
AC characteristics switching register	ACSWR	32	H'FFFD8808	BSC	8, 16, 32
Instruction register	SDIR	16	H'FFFD9000	H-UDI	16
Interrupt control register 0	ICR0	16	H'FFFD9400	INTC	16, 32
Interrupt control register 1	ICR1	16	H'FFFD9402		16, 32
Interrupt control register 2	ICR2	16	H'FFFD9404		16, 32
IRQ interrupt request register	IRQRR	16	H'FFFD9406		16, 32
PINT interrupt enable register	PINTER	16	H'FFFD9408		16, 32
PINT interrupt request register	PIRR	16	H'FFFD940A		16, 32
Bank control register	IBCR	16	H'FFFD940C		16, 32
Bank number register	IBNR	16	H'FFFD940E		16, 32
Interrupt priority register 01	IPR01	16	H'FFFD9418		16, 32
Interrupt priority register 02	IPR02	16	H'FFFD941A		16, 32
Interrupt priority register 05	IPR05	16	H'FFFD9420		16, 32
Interrupt priority register 06	IPR06	16	H'FFFD9800		16, 32
Interrupt priority register 07	IPR07	16	H'FFFD9802		16, 32
Interrupt priority register 08	IPR08	16	H'FFFD9804		16, 32
Interrupt priority register 09	IPR09	16	H'FFFD9806		16, 32
Interrupt priority register 10	IPR10	16	H'FFFD9808		16, 32
Interrupt priority register 11	IPR11	16	H'FFFD980A		16, 32
Interrupt priority register 12	IPR12	16	H'FFFD980C		16, 32
Interrupt priority register 13	IPR13	16	H'FFFD980E		16, 32
Interrupt priority register 14	IPR14	16	H'FFFD9810		16, 32
Interrupt priority register 15	IPR15	16	H'FFFD9812		16, 32
Interrupt priority register 16	IPR16	16	H'FFFD9814		16, 32
Watchdog timer control/status register	WTCSR	16	H'FFFE0000	WDT	16
Watchdog timer counter	WTCNT	16	H'FFFE0002		16
Watchdog reset control/status register	WRCSR	16	H'FFFE0004		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Frequency control register	FRQCR	16	H'FFFE0010	CPG	16
Standby control register	STBCR	8	H'FFFE0014	SYSTEM	8
Standby control register 2	STBCR2	8	H'FFFE0018		8
System control register 1	SYSCR1	8	H'FFFE0402		8
System control register 2	SYSCR2	8	H'FFFE0404		8
Standby control register 3	STBCR3	8	H'FFFE0408		8
Standby control register 4	STBCR4	8	H'FFFE040C		8
Standby control register 5	STBCR5	8	H'FFFE0410		8
64-Hz counter	R64CNT	8	H'FFFE0800	RTC	8
Second counter	RSECCNT	8	H'FFFE0802		8
Minute counter	RMINCNT	8	H'FFFE0804		8
Hour counter	RHRCNT	8	H'FFFE0806		8
Day of week counter	RWKCNT	8	H'FFFE0808		8
Date counter	RDAYCNT	8	H'FFFE080A		8
Month counter	RMONCNT	8	H'FFFE080C		8
Year counter	RYRCNT	16	H'FFFE080E		16
Second alarm register	RSECAR	8	H'FFFE0810		8
Minute alarm register	RMINAR	8	H'FFFE0812		8
Hour alarm register	RHRAR	8	H'FFFE0814		8
Day of week alarm register	RWKAR	8	H'FFFE0816		8
Date alarm register	RDAYAR	8	H'FFFE0818		8
Month alarm register	RMONAR	8	H'FFFE081A		8
RTC control register 1	RCR1	8	H'FFFE081C		8
RTC control register 2	RCR2	8	H'FFFE081E		8
Year alarm register	RYRAR	16	H'FFFE0820		16
RTC control register 3	RCR3	8	H'FFFE0824		8
Port A data register H	PADRH	16	H'FFFE3800	I/O ports	8, 16, 32
Port A data register L	PADRL	16	H'FFFE3802		8, 16
Port A port register H	PAPRH	16	H'FFFE3804		8, 16, 32
Port A port register L	PAPRL	16	H'FFFE3806		8, 16
Port B data register H	PBDRH	16	H'FFFE3808		8, 16, 32
Port B data register L	PBDRL	16	H'FFFE380A		8, 16
Port B port register H	PBPRH	16	H'FFFE380C		8, 16, 32
Port B port register L	PBPRL	16	H'FFFE380E		8, 16
Port C data register H	PCDRH	16	H'FFFE3810		8, 16, 32
Port C data register L	PCDRL	16	H'FFFE3812		8, 16, 32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Port C port register H	PCPRH	16	H'FFFE3814	I/O ports	8, 16
Port C port register L	PCPRL	16	H'FFFE3816		8, 16
Port D data register	PDDR	16	H'FFFE381A		8, 16
Port D port register H	PDPRH	16	H'FFFE381C		8, 16, 32
Port D port register L	PDPRL	16	H'FFFE381E		8, 16
Port E port register	PEPR	16	H'FFFE3826		8, 16
Port F data register	PFDR	16	H'FFFE382A		8, 16
Port F port register	PFPR	16	H'FFFE382E		8, 16
Port A I/O register H	PAIORH	16	H'FFFE3880	PFC	8, 16, 32
Port A I/O register L	PAIORL	16	H'FFFE3882		8, 16
Port A control register 8	PACR8	16	H'FFFE3884		8, 16, 32
Port A control register 7	PACR7	16	H'FFFE3886		8, 16
Port A control register 6	PACR6	16	H'FFFE3888		8, 16, 32
Port A control register 5	PACR5	16	H'FFFE388A		8, 16
Port A control register 4	PACR4	16	H'FFFE388C		8, 16, 32
Port A control register 3	PACR3	16	H'FFFE388E		8, 16
Port A control register 2	PACR2	16	H'FFFE3890		8, 16, 32
Port A control register 1	PACR1	16	H'FFFE3892		8, 16
Port B I/O register H	PBIORH	16	H'FFFE3898		8, 16, 32
Port B I/O register L	PBIORL	16	H'FFFE389A		8, 16
Port B control register 8	PBCR8	16	H'FFFE389C		8, 16, 32
Port B control register 7	PBCR7	16	H'FFFE389E		8, 16
Port B control register 6	PBCR6	16	H'FFFE38A0		8, 16, 32
Port B control register 5	PBCR5	16	H'FFFE38A2		8, 16
Port B control register 4	PBCR4	16	H'FFFE38A4		8, 16, 32
Port B control register 3	PBCR3	16	H'FFFE38A6		8, 16
Port B control register 2	PBCR2	16	H'FFFE38A8		8, 16, 32
Port B control register 1	PBCR1	16	H'FFFE38AA		8, 16
Port C I/O register H	PCIORH	16	H'FFFE38B0		8, 16, 32
Port C I/O register L	PCIORL	16	H'FFFE38B2		8, 16
Port C control register 7	PCCR7	16	H'FFFE38B6		8, 16
Port C control register 6	PCCR6	16	H'FFFE38B8		8, 16, 32
Port C control register 5	PCCR5	16	H'FFFE38BA		8, 16
Port C control register 4	PCCR4	16	H'FFFE38BC		8, 16, 32
Port C control register 3	PCCR3	16	H'FFFE38BE		8, 16
Port C control register 2	PCCR2	16	H'FFFE38C0		8, 16, 32

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Port C control register 1	PCCR1	16	H'FFFE38C2	PFC	8, 16
Port D I/O register	PDIOR	16	H'FFFE38CA		8, 16
Port D control register 5	PDCR5	16	H'FFFE38D2		8, 16
Port D control register 4	PDCR4	16	H'FFFE38D4		8, 16, 32
Port D control register 3	PDCR3	16	H'FFFE38D6		8, 16
Port D control register 2	PDCR2	16	H'FFFE38D8		8, 16, 32
Port D control register 1	PDCR1	16	H'FFFE38DA		8, 16
Port E control register 2	PECR2	16	H'FFFE38F0		8, 16, 32
Port E control register 1	PECR1	16	H'FFFE38F2		8, 16
Port F I/O register	PFIOR	16	H'FFFE38FA		8, 16
Port F control register 2	PFCR2	16	H'FFFE3908		8, 16, 32
Port F control register 1	PFCR1	16	H'FFFE390A		8, 16
Timer control register_3	TCR_3	8	H'FFFE4200	MTU2	8
Timer control register_4	TCR_4	8	H'FFFE4201		8
Timer mode register_3	TMDR_3	8	H'FFFE4202		8
Timer mode register_4	TMDR_4	8	H'FFFE4203		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 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_3	TIER_3	8	H'FFFE4208		8
Timer interrupt enable register_4	TIER_4	8	H'FFFE4209		8
Timer output master enable register	TOER	8	H'FFFE420A		8
Timer gate control register	TGCR	8	H'FFFE420D		8
Timer output control register 1	TOCR1	8	H'FFFE420E		8
Timer output control register 2	TOCR2	8	H'FFFE420F		8
Timer counter_3	TCNT_3	16	H'FFFE4210		16
Timer counter_4	TCNT_4	16	H'FFFE4212		16
Timer cycle data register	TCDR	16	H'FFFE4214		16
Timer dead time data register	TDDR	16	H'FFFE4216		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 A_4	TGRA_4	16	H'FFFE421C		16
Timer general register B_4	TGRB_4	16	H'FFFE421E		16
Timer subcounter	TCNTS	16	H'FFFE4220		16
Timer cycle buffer register	TCBR	16	H'FFFE4222		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Timer general register C_3	TGRC_3	16	H'FFFE4224	MTU2	16
Timer general register D_3	TGRD_3	16	H'FFFE4226		16
Timer general register C_4	TGRC_4	16	H'FFFE4228		16
Timer general register D_4	TGRD_4	16	H'FFFE422A		16
Timer status register_3	TSR_3	8	H'FFFE422C		8
Timer status register_4	TSR_4	8	H'FFFE422D		8
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 output level buffer register	TOLBR	8	H'FFFE4236		8
Timer buffer operation transfer mode register_3	TBTM_3	8	H'FFFE4238		8
Timer buffer operation transfer mode register_4	TBTM_4	8	H'FFFE4239		8
Timer A/D converter start request control register	TADCR	16	H'FFFE4240		16
Timer A/D converter start request cycle set register A_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 waveform control register	TWCR	8	H'FFFE4260		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 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

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Timer general register D_0	TGRD_0	16	H'FFFE430E	MTU2	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 register 2_0	TIER2_0	8	H'FFFE4324		8
Timer status register 2_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
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 counter U_5	TCNTU_5	16	H'FFFE4080		16
Timer general register U_5	TGRU_5	16	H'FFFE4082		16
Timer control register U_5	TCRU_5	8	H'FFFE4084		8
Timer I/O control register U_5	TIORU_5	8	H'FFFE4086		8
Timer counter V_5	TCNTV_5	16	H'FFFE4090		16
Timer general register V_5	TGRV_5	16	H'FFFE4092		16
Timer control register V_5	TCRV_5	8	H'FFFE4094		8
Timer I/O control register V_5	TIORV_5	8	H'FFFE4096		8
Timer counter W_5	TCNTW_5	8	H'FFFE40A0		8
Timer general register W_5	TGRW_5	16	H'FFFE40A2		16
Timer control register W_5	TCRW_5	8	H'FFFE40A4		8
Timer I/O control register W_5	TIORW_5	8	H'FFFE40A6		8
Timer status register_5	TSR_5	8	H'FFFE40B0		8

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Timer interrupt enable register_5	TIER_5	8	H'FFFE40B2	MTU2	8
Timer start register_5	TSTR_5	8	H'FFFE40B4		8
Timer compare match clear register	TCNTCMPCLR	8	H'FFFE40B6		8
Timer control register_0	T8TCR_0	8	H'FFFE5400	TMR	8
Timer control register_1	T8TCR_1	8	H'FFFE5401		8
Timer control/status register_0	T8TCSR_0	8	H'FFFE5402		8
Timer control/status register_1	T8TCSR_1	8	H'FFFE5403		8
Time constant register A_0	T8TCORA_0	8	H'FFFE5404		8
Time constant register A_1	T8TCORA_1	8	H'FFFE5405		8
Time constant register B_0	T8TCORB_0	8	H'FFFE5406		8
Time constant register B_1	T8TCORB_1	8	H'FFFE5407		8
Timer counter_0	T8TCNT_0	8	H'FFFE5408		8
Timer counter_1	T8TCNT_1	8	H'FFFE5409		8
Timer counter control register_0	T8TCCR_0	8	H'FFFE540A		8
Timer counter control register_1	T8TCCR_1	8	H'FFFE540B		8
A/D data register A_0	ADDRA	16	H'FFFE5800	ADC	16
A/D data register B_0	ADDRB	16	H'FFFE5802		16
A/D data register C_0	ADDRC	16	H'FFFE5804		16
A/D data register D_0	ADDRD	16	H'FFFE5806		16
A/D data register E_0	ADDRE	16	H'FFFE5808		16
A/D data register F_0	ADDRF	16	H'FFFE580A		16
A/D data register G_0	ADDRG	16	H'FFFE580C		16
A/D data register H_0	ADDRH	16	H'FFFE580E		16
A/D control/status register	ADCSR	16	H'FFFE5820		16
D/A data register 0	DADR0	8	H'FFFE6800	DAC	8, 16
D/A data register 1	DADR1	8	H'FFFE6801		8, 16
D/A control register	DACR	8	H'FFFE6802		8, 16
Serial mode register_0	SCSMR_0	16	H'FFFE8000	SCIF	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	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

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Line status register_0	SCLSR_0	16	H'FFFE8024	SCIF	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 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
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 mode register_3	SCSMR_3	16	H'FFFE9800		16
Bit rate register_3	SCBRR_3	8	H'FFFE9804		8
Serial control register_3	SCSCR_3	16	H'FFFE9808		16
Transmit FIFO data register_3	SCFTDR_3	8	H'FFFE980C		8
Serial status register_3	SCFSR_3	16	H'FFFE9810		16
Receive FIFO data register_3	SCFRDR_3	8	H'FFFE9814		8
FIFO control register_3	SCFCR_3	16	H'FFFE9818		16
FIFO data count register_3	SCFDR_3	16	H'FFFE981C		16
Serial port register_3	SCSPTR_3	16	H'FFFE9820		16
Line status register_3	SCLSR_3	16	H'FFFE9824		16
Serial mode register_4	SCSMR_4	16	H'FFFEA000		16
Bit rate register_4	SCBRR_4	8	H'FFFEA004		8
Serial control register_4	SCSCR_4	16	H'FFFEA008		16
Transmit FIFO data register_4	SCFTDR_4	8	H'FFFEA00C		8
Serial status register_4	SCFSR_4	16	H'FFFEA010		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Receive FIFO data register_4	SCFRDR_4	8	H'FFFEA014	SCIF	8
FIFO control register_4	SCFCR_4	16	H'FFFEA018		16
FIFO data count register_4	SCFDR_4	16	H'FFFEA01C		16
Serial port register_4	SCSPTR_4	16	H'FFFEA020		16
Line status register_4	SCLSR_4	16	H'FFFEA024		16
Serial mode register_5	SCSMR_5	16	H'FFFEA800		16
Bit rate register_5	SCBRR_5	8	H'FFFEA804		8
Serial control register_5	SCSCR_5	16	H'FFFEA808		16
Transmit FIFO data register_5	SCFTDR_5	8	H'FFFEA80C		8
Serial status register_5	SCFSR_5	16	H'FFFEA810		16
Receive FIFO data register_5	SCFRDR_5	8	H'FFFEA814		8
FIFO control register_5	SCFCR_5	16	H'FFFEA818		16
FIFO data count register_5	SCFDR_5	16	H'FFFEA81C		16
Serial port register_5	SCSPTR_5	16	H'FFFEA820		16
Line status register_5	SCLSR_5	16	H'FFFEA824		16
Serial mode register_6	SCSMR_6	16	H'FFFEB000		16
Bit rate register_6	SCBRR_6	8	H'FFFEB004		8
Serial control register_6	SCSCR_6	16	H'FFFEB008		16
Transmit FIFO data register_6	SCFTDR_6	8	H'FFFEB00C		8
Serial status register_6	SCFSR_6	16	H'FFFEB010		16
Receive FIFO data register_6	SCFRDR_6	8	H'FFFEB014		8
FIFO control register_6	SCFCR_6	16	H'FFFEB018		16
FIFO data count register_6	SCFDR_6	16	H'FFFEB01C		16
Serial port register_6	SCSPTR_6	16	H'FFFEB020		16
Line status register_6	SCLSR_6	16	H'FFFEB024		16
Serial mode register_7	SCSMR_7	16	H'FFFEB800		16
Bit rate register_7	SCBRR_7	8	H'FFFEB804		8
Serial control register_7	SCSCR_7	16	H'FFFEB808		16
Transmit FIFO data register_7	SCFTDR_7	8	H'FFFEB80C		8
Serial status register_7	SCFSR_7	16	H'FFFEB810		16
Receive FIFO data register_7	SCFRDR_7	8	H'FFFEB814		8
FIFO control register_7	SCFCR_7	16	H'FFFEB818		16
FIFO data count register_7	SCFDR_7	16	H'FFFEB81C		16
Serial port register_7	SCSPTR_7	16	H'FFFEB820		16
Line status register_7	SCLSR_7	16	H'FFFEB824		16

Register Name	Abbreviation	Number of Bits	Address	Module	Access Size
Control register_0	SSICR_0	32	H'FFFED000	SSI	32
Status register_0	SSISR_0	32	H'FFFED004		32
Transmit data register_0	SSITDR_0	32	H'FFFED008		32
Receive data register_0	SSIRDR_0	32	H'FFFED00C		32
Control register_1	SSICR_1	32	H'FFFED080		32
Status register_1	SSISR_1	32	H'FFFED084		32
Transmit data register_1	SSITDR_1	32	H'FFFED088		32
Receive data register_1	SSIRDR_1	32	H'FFFED08C		32
I^2C bus control register 1_0	ICCR1_0	8	H'FFFEE000	IIC3	8
I^2C bus control register 2_0	ICCR2_0	8	H'FFFEE001		8
I^2C bus mode register_0	ICMR_0	8	H'FFFEE002		8
I^2C bus interrupt enable register_0	ICIER_0	8	H'FFFEE003		8
I^2C bus status register_0	ICSR_0	8	H'FFFEE004		8
Slave address register_0	SAR_0	8	H'FFFEE005		8
I^2C bus transmit data register_0	ICDRT_0	8	H'FFFEE006		8
I^2C bus receive data register_0	ICDRR_0	8	H'FFFEE007		8
NF2CYC register_0	NF2CYC_0	8	H'FFFEE008		8
I^2C bus control register 1_1	ICCR1_1	8	H'FFFEE080		8
I^2C bus control register 2_1	ICCR2_1	8	H'FFFEE081		8
I^2C bus mode register_1	ICMR_1	8	H'FFFEE082		8
I^2C bus interrupt enable register_1	ICIER_1	8	H'FFFEE083		8
I^2C bus status register_1	ICSR_1	8	H'FFFEE084		8
Slave address register_1	SAR_1	8	H'FFFEE085		8
I^2C bus transmit dataregister_1	ICDRT_1	8	H'FFFEE086		8
I^2C bus receive data register_1	ICDRR_1	8	H'FFFEE087		8
NF2CYC register_1	NF2CYC_1	8	H'FFFEE088		8
I^2C bus control register 1_2	ICCR1_2	8	H'FFFEE100		8
I^2C bus control register 2_2	ICCR2_2	8	H'FFFEE101		8
I^2C bus mode register_2	ICMR_2	8	H'FFFEE102		8
I^2C bus interrupt enable register_2	ICIER_2	8	H'FFFEE103		8
I^2C bus status register_2	ICSR_2	8	H'FFFEE104		8
Slave address register_2	SAR_2	8	H'FFFEE105		8
I^2C bus transmit data register_2	ICDRT_2	8	H'FFFEE106		8
I^2C bus receive data register_2	ICDRR_2	8	H'FFFEE107		8
NF2CYC register_2	NF2CYC_2	8	H'FFFEE108		8

Register Name		Abbreviation	Number of Bits	Address	Module	Access Size
Master control register_0		MCR_0	16	H'FFFF0000	RCAN-ET	16
General status register_0		GSR_0	16	H'FFFF0002		16
Bit configuration register 1_0		BCR1_0	16	H'FFFF0004		16
Bit configuration register 0_0		BCR0_0	16	H'FFFF0006		16
Interrupt request register_0		IRR_0	16	H'FFFF0008		16
Interrupt mask register_0		IMR_0	16	H'FFFF000A		16
Transmit error counter_0/ Receive error counter_0		TEC_0/ REC_0	16	H'FFFF000C		16
Transmit pending register 1_0		TXPR1_0	16	H'FFFF0020		32
Transmit pending register 0_0		TXPR0_0	16	H'FFFF0022		
Transmit cancel register 0_0		TXCR0_0	16	H'FFFF002A		16
Transmit acknowledge register 0_0		TXACK0_0	16	H'FFFF0032		16
Abort acknowledge register 0_0		ABACK0_0	16	H'FFFF003A		16
Data frame receive pending register 0_0		RXPR0_0	16	H'FFFF0042		16
Remote frame receive pending register 0_0		RFPR0_0	16	H'FFFF004A		16
Mailbox interrupt mask register0_0		MBIMR0_0	16	H'FFFF0052		16
Unread message status register 0_0		UMSR0_0	16	H'FFFF005A		16
Mailbox 0	Control 0	CONTROL0H	16	H'FFFF0100		16, 32
		CONTROL0L	16	H'FFFF0102		16
	LAFM	LAFMH	16	H'FFFF0104		16, 32
		LAFML	16	H'FFFF0106		16
	Data	MSG_DATA[0]	8	H'FFFF0108		8, 16, 32
		MSG_DATA[1]	8	H'FFFF0109		8
		MSG_DATA[2]	8	H'FFFF010A		8, 16
		MSG_DATA[3]	8	H'FFFF010B		8
		MSG_DATA[4]	8	H'FFFF010C		8, 16, 32
		MSG_DATA[5]	8	H'FFFF010D		8
		MSG_DATA[6]	8	H'FFFF010E		8, 16
		MSG_DATA[7]	8	H'FFFF010F		8
	Control 1	CONTROL1H	8	H'FFFF0110		8, 16
		CONTROL1L	8	H'FFFF0111		8

Register Name		Abbreviation	Number of Bits	Address	Module	Access Size
Mailbox n (n = 1 to 15)	Control 0	CONTROL0H	16	H'FFFF0100 + n × 32	RCAN-ET	16, 32
		CONTROL0L	16	H'FFFF0102 + n × 32		16
	LAFM	LAFMH	16	H'FFFF0104 + n × 32		16, 32
		LAFML	16	H'FFFF0106 + n × 32		16
	Data	MSG_DATA[0]	8	H'FFFF0108 + n × 32		8, 16, 32
		MSG_DATA[1]	8	H'FFFF0109 + n × 32		8
		MSG_DATA[2]	8	H'FFFF010A + n × 32		8, 16
Mailbox n (n = 1 to 15)	Data	MSG_DATA[3]	8	H'FFFF010B + n × 32		8
		MSG_DATA[4]	8	H'FFFF010C + n × 32		8, 16, 32
		MSG_DATA[5]	8	H'FFFF010D + n × 32		8
		MSG_DATA[6]	8	H'FFFF010E + n × 32		8, 16
		MSG_DATA[7]	8	H'FFFF010F + n × 32		8
	Control 1	CONTROL1H	8	H'FFFF0110 + n × 32		8, 16
		CONTROL1L	8	H'FFFF0111 + n × 32		8
Master control register_1		MCR_1	16	H'FFFF0800		16
General status register_1		GSR_1	16	H'FFFF0802		16
Bit configuration register1_1		BCR1_1	16	H'FFFF0804		16
Bit configuration register0_1		BCR0_1	16	H'FFFF0806		16
Interrupt request register_1		IRR_1	16	H'FFFF0808		16
Interrupt mask register_1		IMR_1	16	H'FFFF080A		16
Transmit error counter_1/ Receive error counter_1		TEC_1/ REC_1	16	H'FFFF080C		16
Transmit pending register 1_1		TXPR1_1	16	H'FFFF0820		32
Transmit pending register 0_1		TXPR0_1	16	H'FFFF0822		
Transmit cancel register 0_1		TXCR0_1	16	H'FFFF082A		16
Transmit acknowledge register 0_1		TXACK0_1	16	H'FFFF0832		16

Register Name		Abbreviation	Number of Bits	Address	Module	Access Size
Abort acknowledge register 0_1		ABACK0_1	16	H'FFFF083A	RCAN-ET	16
Data frame receive pending register 0_1		RXPR0_1	16	H'FFFF0842		16
Remote frame receive pending register 0_1		RFPR0_1	16	H'FFFF084A		16
Mailbox interrupt mask register0_1		MBIMR0_1	16	H'FFFF0852		16
Unread message status register 0_1		UMSR0_1	16	H'FFFF085A		16
Mailbox 0	Control 0	CONTROL0H	16	H'FFFF0900		16, 32
		CONTROL0L	16	H'FFFF0902		16
	LAFM	LAFMH	16	H'FFFF0904		16, 32
		LAFML	16	H'FFFF0906		16
	Data	MSG_DATA[0]	8	H'FFFF0908		8, 16, 32
		MSG_DATA[1]	8	H'FFFF0909		8
		MSG_DATA[2]	8	H'FFFF090A		8, 16
		MSG_DATA[3]	8	H'FFFF090B		8
		MSG_DATA[4]	8	H'FFFF090C		8, 16, 32
		MSG_DATA[5]	8	H'FFFF090D		8
		MSG_DATA[6]	8	H'FFFF090E		8, 16
		MSG_DATA[7]	8	H'FFFF090F		8
	Control 1	CONTROL1H	8	H'FFFF0910		8, 16
		CONTROL1L	8	H'FFFF0911		8
Mailbox n (n = 1 to 15)	Control 0	CONTROL0H	16	H'FFFF0900 + n × 32		16, 32
		CONTROL0L	16	H'FFFF0902 + n × 32		16
	LAFM	LAFMH	16	H'FFFF0904 + n × 32		16, 32
		LAFML	16	H'FFFF0906 + n × 32		16
	Data	MSG_DATA[0]	8	H'FFFF0908•n× 32		8, 16, 32
		MSG_DATA[1]	8	H'FFFF0909•n× 32		8
		MSG_DATA[2]	8	H'FFFF090A + n × 32		8, 16
		MSG_DATA[3]	8	H'FFFF090B + n × 32		8
		MSG_DATA[4]	8	H'FFFF090C + n × 32		8, 16, 32

Register Name		Abbreviation	Number of Bits	Address	Module	Access Size
Mailbox n (n = 1 to 15)	Data	MSG_DATA[5]	8	H'FFFF090D + n × 32	RCAN-ET	8
		MSG_DATA[6]	8	H'FFFF090E + n × 32		8, 16
		MSG_DATA[7]	8	H'FFFF090F + n × 32		8
	Control 1	CONTROL1H	8	H'FFFF0910 + n × 32		8, 16
		CONTROL1L	8	H'FFFF0911 + n × 32		8
DMA transfer enable register 0		DREQER0	8	H'FFFF1600	INTC	8, 16, 32
DMA transfer enable register 1		DREQER1	8	H'FFFF1601		8
DMA transfer enable register 2		DREQER2	8	H'FFFF1602		8, 16
DMA transfer enable register 3		DREQER3	8	H'FFFF1603		8
Deep standby cancel source flag register		DSFR	16	H'FFFF1904	SYSTEM	16
Deep standby oscillation stabilization clock select register		DSCNT	8	H'FFFF1906		8
RAM retained area specification register		RAMKP	8	H'FFFF1907		8

28.2 Register Bits

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

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

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SYCBEEN	STSCLR	—	—	—	—	—	—	—	Bus Monitor
	—	—	—	—	—	TOEN	IGAEN	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SYCBESTS1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	PTO	PER	—	—	—	PMST1	PMST0	
	—	—	—	—	—	—	—	—	
SYCBESTS2	—	ETO	EER	—	—	—	EMST1	EMST0	
	—	—	—	—	—	—	—	—	
	—	—	OER	—	—	—	OMST1	OMST0	
	—	—	SHER	—	—	—	SHMST1	SHMST0	
SYCBESW	00CPEN	—	10CPEN	11CPEN	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS0CNT	—	—	—	—	—	—	—	—	BSC
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS0REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
CS2CNT	—	—	—	—	—	—	—	—	BSC
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS2REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS3CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS3REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS4CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS4REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS5CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS5REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS6CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
CS6REC	—	—	—	—	WRCV3	WRCV2	WRCV1	WRCV0	BSC
	—	—	—	—	RRCV3	RRCV2	RRCV1	RRCV0	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SDC0CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
SDC1CNT	—	—	—	—	—	—	—	—	
	—	—	BSIZE1	BSIZE0	—	—	—	EXENB	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CSMOD0	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT0	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT0	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	
CSMOD1	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT1	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT1	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
CSMOD2	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	BSC
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT2	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT2	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	
CSMOD3	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT3	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT3	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	
CSMOD4	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT4	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT4	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
CSMOD5	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	BSC
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT5	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT5	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	
CSMOD6	PRMOD	—	PBCNT1	PBCNT0	—	—	PWENB	PRENB	
	—	—	—	—	EWENB	—	—	WRMOD	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CS1WCNT6	—	—	—	CSRWAIT4	CSRWAIT3	CSRWAIT2	CSRWAIT1	CSRWAIT0	
	—	—	—	CSWWAIT4	CSWWAIT3	CSWWAIT2	CSWWAIT1	CSWWAIT0	
	—	—	—	—	—	CSPRWAIT2	CSPRWAIT1	CSPRWAIT0	
	—	—	—	—	—	CSPWWAIT2	CSPWWAIT1	CSPWWAIT0	
CS2WCNT6	—	CSON2	CSON1	CSON0	—	WDON2	WDON1	WDON0	
	—	WRON2	WRON1	WRON0	—	RDON2	RDON1	RDON0	
	—	—	—	—	—	WDOFF2	WDOFF1	WDOFF0	
	—	CSWOFF2	CSWOFF1	CSWOFF0	—	CSROFF2	CSROFF1	CSROFF0	
SDRFCNT0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DSFEN	
SDRFCNT1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DRFEN	
	DREFW3	DREFW2	DREFW1	DREFW0	DRFC11	DRFC10	DRFC9	DRFC8	
	DRFC7	DRFC6	DRFC5	DRFC4	DRFC3	DRFC2	DRFC1	DRFC0	
SDIR0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	DPC2	DPC1	DPC0	
	DARFC3	DARFC2	DARFC1	DARFC0	DARFI3	DARFI2	DARFI1	DARFI0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SDIR1	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	DINIST	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DINIRQ	
SDPWDCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DPWD	
SDDPWDCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DDPD	
SD0ADR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	DDBW1	DDBW0	
	—	—	—	—	—	DSZ2	DSZ1	DSZ0	
SD0TR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	DRAS2	DRAS1	DRAS0	
	—	—	DRCD1	DRCD0	DPCG2	DPCG1	DPCG0	DWR	
	—	—	—	—	—	DCL2	DCL1	DCL0	
SD0MOD	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	DMR14	DMR13	DMR12	DMR11	DMR10	DMR9	DMR8	
	DMR7	DMR6	DMR5	DMR4	DMR3	DMR2	DMR1	DMR0	
SD1ADR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	DDBW1	DDBW0	
	—	—	—	—	—	DSZ2	DSZ1	DSZ0	
SD1TR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	DRAS2	DRAS1	DRAS0	
	—	—	DRCD1	DRCD0	DPCG2	DPCG1	DPCG0	DWR	
	—	—	—	—	—	DCL2	DCL1	DCL0	
SD1MOD	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	DMR14	DMR13	DMR12	DMR11	DMR10	DMR9	DMR8	
	DMR7	DMR6	DMR5	DMR4	DMR3	DMR2	DMR1	DMR0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SDSTR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	DSRFST	DINIST	DPWDST	DDPDST	DMRSST	
SDCKSCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DCKSEN	
	—	—	—	—	—	—	—	—	
	DCKSC7	DCKSC6	DCKSC5	DCKSC4	DCKSC3	DCKSC2	DCKSC1	DCKSC0	
DMCSADR0	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	DMAC
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR0	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT0	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD0	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMCSADR1	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR1	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT1	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMMOD1	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	DMAC
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMCSADR2	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR2	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT2	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD2	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMCSADR3	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR3	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT3	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD3	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMCSADR4	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	DMAC
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR4	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT4	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD4	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMCSADR5	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR5	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT5	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD5	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMCSADR6	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMCDADR6	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	DMAC
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT6	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD6	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMCSADR7	CSA31	CSA30	CSA29	CSA28	CSA27	CSA26	CSA25	CSA24	
	CSA23	CSA22	CSA21	CSA20	CSA19	CSA18	CSA17	CSA16	
	CSA15	CSA14	CSA13	CSA12	CSA11	CSA10	CSA9	CSA8	
	CSA7	CSA6	CSA5	CSA4	CSA3	CSA2	CSA1	CSA0	
DMCDADR7	CDA31	CDA30	CDA29	CDA28	CDA27	CDA26	CDA25	CDA24	
	CDA23	CDA22	CDA21	CDA20	CDA19	CDA18	CDA17	CDA16	
	CDA15	CDA14	CDA13	CDA12	CDA11	CDA10	CDA9	CDA8	
	CDA7	CDA6	CDA5	CDA4	CDA3	CDA2	CDA1	CDA0	
DMCBCT7	—	—	—	—	—	—	CBC25	CBC24	
	CBC23	CBC22	CBC21	CBC20	CBC19	CBC18	CBC17	CBC16	
	CBC15	CBC14	CBC13	CBC12	CBC11	CBC10	CBC9	CBC8	
	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1	CBC0	
DMMOD7	—	—	—	—	OPSEL3	OPSEL2	OPSEL1	OPSEL0	
	—	—	—	—	—	SZSEL2	SZSEL1	SZSEL0	
	—	SAMOD2	SAMOD1	SAMOD0	—	DAMOD2	DAMOD1	DAMOD0	
	—	—	—	—	SACT	DACT	DTCM1	DTCM0	
DMRSADR0	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR0	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMRBCT0	—	—	—	—	—	—	RBC25	RBC24	DMAC
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR1	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR1	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	
DMRBCT1	—	—	—	—	—	—	RBC25	RBC24	
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR2	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR2	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	
DMRBCT2	—	—	—	—	—	—	RBC25	RBC24	
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR3	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR3	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMRBCT3	—	—	—	—	—	—	RBC25	RBC24	DMAC
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR4	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR4	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	
DMRBCT4	—	—	—	—	—	—	RBC25	RBC24	
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR5	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR5	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	
DMRBCT5	—	—	—	—	—	—	RBC25	RBC24	
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR6	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR6	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMRBCT6	—	—	—	—	—	—	RBC25	RBC24	DMAC
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMRSADR7	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	
	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16	
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	
	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0	
DMRDADR7	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	
	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16	
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	
	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0	
DMRBCT7	—	—	—	—	—	—	RBC25	RBC24	
	RBC23	RBC22	RBC21	RBC20	RBC19	RBC18	RBC17	RBC16	
	RBC15	RBC14	RBC13	RBC12	RBC11	RBC10	RBC9	RBC8	
	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	
DMCNTA0	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB0	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMCNTA1	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB1	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMCNTA2	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMCNTB2	—	—	—	—	—	—	—	DEN	DMAC
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMCNTA3	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB3	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMCNTA4	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB4	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMCNTA5	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB5	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMCNTA6	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB6	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
DMCNTA7	—	—	MDSEL1	MDSEL0	—	—	DSEL1	DSEL0	DMAC
	—	—	—	—	—	—	STRG1	STRG0	
	—	—	—	—	—	BRLOD	SRLOD	DRLOD	
	—	—	DCTG5	DCTG4	DCTG3	DCTG2	DCTG1	DCTG0	
DMCNTB7	—	—	—	—	—	—	—	DEN	
	—	—	—	—	—	—	—	DREQ	
	—	—	—	—	—	—	—	ECLR	
	—	—	—	—	—	—	—	DSCLR	
DMSCNT	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	DMST	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMICNT	DINTM_CH0	DINTM_CH1	DINTM_CH2	DINTM_CH3	DINTM_CH4	DINTM_CH5	DINTM_CH6	DINTM_CH7	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMICNTA	DINTA_CH0	DINTA_CH1	DINTA_CH2	DINTA_CH3	DINTA_CH4	DINTA_CH5	DINTA_CH6	DINTA_CH7	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMISTS	DISTS_CH0	DISTS_CH1	DISTS_CH2	DISTS_CH3	DISTS_CH4	DISTS_CH5	DISTS_CH6	DISTS_CH7	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMEDET	DEDET_CH0	DEDET_CH1	DEDET_CH2	DEDET_CH3	DEDET_CH4	DEDET_CH5	DEDET_CH6	DEDET_CH7	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
DMASTS	DASTS_CH0	DASTS_CH1	DASTS_CH2	DASTS_CH3	DASTS_CH4	DASTS_CH5	DASTS_CH6	DASTS_CH7	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
BAR_0	BA0_31	BA0_30	BA0_29	BA0_28	BA0_27	BA0_26	BA0_25	BA0_24	UBC
	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	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
BAMR_0	BAM0_31	BAM0_30	BAM0_29	BAM0_28	BAM0_27	BAM0_26	BAM0_25	BAM0_24	UBC
	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	
BDR_0	BD0_31	BD0_30	BD0_29	BD0_28	BD0_27	BD0_26	BD0_25	BD0_24	
	BD0_23	BD0_22	BD0_21	BD0_20	BD0_19	BD0_18	BD0_17	BD0_16	
	BD0_15	BD0_14	BD0_13	BD0_12	BD0_11	BD0_10	BD0_9	BD0_8	
	BD0_7	BD0_6	BD0_5	BD0_4	BD0_3	BD0_2	BD0_1	BD0_0	
BDMR_0	BDM0_31	BDM0_30	BDM0_29	BDM0_28	BDM0_27	BDM0_26	BDM0_25	BDM0_24	
	BDM0_23	BDM0_22	BDM0_21	BDM0_20	BDM0_19	BDM0_18	BDM0_17	BDM0_16	
	BDM0_15	BDM0_14	BDM0_13	BDM0_12	BDM0_11	BDM0_10	BDM0_9	BDM0_8	
	BDM0_7	BDM0_6	BDM0_5	BDM0_4	BDM0_3	BDM0_2	BDM0_1	BDM0_0	
BAR_1	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_1	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	
BDR_1	BD0_31	BD0_30	BD0_29	BD0_28	BD0_27	BD0_26	BD0_25	BD0_24	
	BD0_23	BD0_22	BD0_21	BD0_20	BD0_19	BD0_18	BD0_17	BD0_16	
	BD0_15	BD0_14	BD0_13	BD0_12	BD0_11	BD0_10	BD0_9	BD0_8	
	BD0_7	BD0_6	BD0_5	BD0_4	BD0_3	BD0_2	BD0_1	BD0_0	
BDMR_1	BDM0_31	BDM0_30	BDM0_29	BDM0_28	BDM0_27	BDM0_26	BDM0_25	BDM0_24	
	BDM0_23	BDM0_22	BDM0_21	BDM0_20	BDM0_19	BDM0_18	BDM0_17	BDM0_16	
	BDM0_15	BDM0_14	BDM0_13	BDM0_12	BDM0_11	BDM0_10	BDM0_9	BDM0_8	
	BDM0_7	BDM0_6	BDM0_5	BDM0_4	BDM0_3	BDM0_2	BDM0_1	BDM0_0	
BBR_0	—	—	UBID0	DBE0	—	—	CP0_1	CP0_0	
	CD0_1	CD0_0	ID0_1	ID0_0	RW0_1	RW0_0	SZ0_1	SZ0_0	
BBR_1	—	—	UBID1	DBE1	—	—	CP1_1	CP1_0	
	CD1_1	CD1_0	ID1_1	ID1_0	RW1_1	RW1_0	SZ1_1	SZ1_0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	CKS1	CKS2	
	SCMFC0	SCMFC1	SCMFD0	SCMFD1	—	—	—	BDI	
	—	PCB1	PCB0	—	—	—	—	—	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
CCR1	—	—	—	—	—	—	—	—	Cache
	—	—	—	—	—	—	—	—	
	—	—	—	—	ICF	—	—	ICE	
	—	—	—	—	OCF	—	WT	OCE	
CCR2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	LE	
	—	—	—	—	—	—	W3LOAD	W3LOCK	
	—	—	—	—	—	—	W2LOAD	W2LOCK	
ACSWR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	ACOSW3	ACOSW2	ACOSW1	ACOSW0	
SDIR	TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	H-UDI
	—	—	—	—	—	—	—	—	
ICR0	NMIL	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
ICR2	—	—	—	—	—	—	—	—	
	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S	
IRQRR	—	—	—	—	—	—	—	—	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
PINTER	—	—	—	—	—	—	—	—	
	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E	
PIRR	—	—	—	—	—	—	—	—	
	PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R	
IBCR	E15	E14	E13	E12	E11	E10	E9	E8	
	E7	E6	E5	E4	E3	E2	E1	—	
IBNR	BE1	BE0	BOVE	—	—	—	—	—	
	—	—	—	—	BN3	BN2	BN1	BN0	
IPR01	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR02	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR05	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
IPR06	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	INTC
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR07	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR08	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR09	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR10	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR11	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR12	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR13	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR14	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR15	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
IPR16	IP33	IP32	IP31	IP30	IP23	IP22	IP21	IP20	
	IP13	IP12	IP11	IP10	IP03	IP02	IP01	IP00	
WTCSR	IOVF	WTIT	TME	—	—	CKS2	CKS1	CKS0	WDT
	—	—	—	—	—	—	—	—	
WTCNT	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0	
	—	—	—	—	—	—	—	—	
WRCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
FRQCR	—	—	—	CKOEN	—	STC2	STC1	STC0	CPG
	—	IFC2	IFC1	IFC0	RNGS	PFC2	PFC1	PFC0	
STBCR	STBY	DEEP	—	—	—	—	MSTP1	—	SYSTEM
STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	
SYSCR1	—	—	—	—	—	—	RAME1	RAME0	
SYSCR2	—	—	—	—	—	—	RAMWE1	RAMWE0	
STBCR3	—	—	MSTP35	—	MSTP33	MSTP32	MSTP31	—	
STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40	
STBCR5	MSTP57	MSTP56	MSTP55	—	MSTP53	MSTP52	—	CKDV3	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz	RTC
RSECCNT	—	10 seconds					1second		
RMINCNT	—	10 minutes					1 minute		
RHRCNT	—	—	10 hours				1 hour		
RWKCNT	—	—	—	—	—		Day		
RDAYCNT	—	—	10 days				1 day		
RMONCNT	—	—	—	10 months			1 month		
RYRCNT	1000 years						100 years		
	10 years						1 year		
RSECAR	ENB	10 seconds					1 second		
RMINAR	ENB	10 minutes					1 minute		
RHRAR	ENB	—	10 hours				1 hour		
RWKAR	ENB	—	—	—	—	Day			
RDAYAR	ENB	—	10 days				1 day		
RMONAR	ENB	—	—	10 months			1 month		
RCR1	CF	—	—	CIE	AIE	—	—	AF	
RCR2	PEF	PES2	PES1	PES0	RTCEN	ADJ	RESET	START	
RYRAR	1000 years						100 years		
	10 years						1 year		
RCR3	ENB	—	—	—	—	—	—	—	
PADRH	PA31DR	PA30DR	PA29DR	PA28DR	PA27DR	PA26DR	PA25DR	PA24DR	I/O ports
	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	PA31PR	PA30PR	PA29PR	PA28PR	PA27PR	PA26PR	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	PB31DR	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	PB31PR	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	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
PCDRH	—	—	—	—	—	—	—	—	I/O ports
	—	—	PC21DR	PC20DR	PC19DR	PC18DR	PC17DR	PC16DR	
PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PCPRH	—	—	—	—	—	—	PC25PR	PC24PR	
	PC23PR	PC22PR	PC21PR	PC20PR	PC19PR	PC18PR	PC17PR	PC16PR	
PCPRL	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	
	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR	
PDDR	—	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PDPRH	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	PD16PR	
PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	
	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR	
PEPR	—	—	—	—	—	—	—	—	
	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR	
PFDR	—	—	—	—	—	—	—	—	
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PFPR	—	—	—	—	—	—	—	—	
	PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR	
PAIORH	PA31IOR	PA30IOR	PA29IOR	PA28IOR	PA27IOR	PA26IOR	PA25IOR	PA24IOR	PFC
	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	
PACR8	—	—	PA31MD1	PA31MD0	—	—	PA30MD1	PA30MD0	
	—	—	PA29MD1	PA29MD0	—	—	PA28MD1	PA28MD0	
PACR7	—	PA27MD2	PA27MD1	PA27MD0	—	PA26MD2	PA26MD1	PA26MD0	
	—	PA25MD2	PA25MD1	PA25MD0	—	—	PA24MD1	PA24MD0	
PACR6	—	—	—	PA23MD0	—	—	—	PA22MD0	
	—	—	—	PA21MD0	—	—	—	PA20MD0	
PACR5	—	—	—	PA19MD0	—	—	—	PA18MD0	
	—	—	—	PA17MD0	—	—	—	PA16MD0	
PACR4	—	—	—	PA15MD0	—	—	—	PA14MD0	
	—	—	—	PA13MD0	—	—	—	PA12MD0	
PACR3	—	—	—	PA11MD0	—	—	—	PA10MD0	
	—	—	—	PA9MD0	—	—	—	PA8MD0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
PACR2	—	—	—	PA7MD0	—	—	—	PA6MD0	PFC
	—	—	—	PA5MD0	—	—	—	PA4MD0	
PACR1	—	—	—	PA3MD0	—	—	—	PA2MD0	
	—	—	—	PA1MD0	—	—	—	PA0MD0	
PBIORH	PB31IOR	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	
PBCR8	—	—	PB31MD1	PB31MD0	—	PB30MD2	PB30MD1	PB30MD0	
	—	PB29MD2	PB29MD1	PB29MD0	—	PB28MD2	PB28MD1	PB28MD0	
PBCR7	—	—	PB27MD1	PB27MD0	—	PB26MD2	PB26MD1	PB26MD0	
	—	PB25MD2	PB25MD1	PB25MD0	—	PB24MD2	PB24MD1	PB24MD0	
PBCR6	—	—	PB23MD1	PB23MD0	—	PB22MD2	PB22MD1	PB22MD0	
	—	PB21MD2	PB21MD1	PB21MD0	—	PB20MD2	PB20MD1	PB20MD0	
PBCR5	—	—	PB19MD1	PB19MD0	—	—	PB18MD1	PB18MD0	
	—	—	PB17MD1	PB17MD0	—	—	PB16MD1	PB16MD0	
PBCR4	—	—	—	PB15MD0	—	—	—	PB14MD0	
	—	—	—	PB13MD0	—	—	—	PB12MD0	
PBCR3	—	—	—	PB11MD0	—	—	—	PB10MD0	
	—	—	—	PB9MD0	—	—	—	PB8MD0	
PBCR2	—	—	—	PB7MD0	—	—	—	PB6MD0	
	—	—	—	PB5MD0	—	—	—	PB4MD0	
PBCR1	—	—	—	PB3MD0	—	—	—	PB2MD0	
	—	—	—	PB1MD0	—	—	—	PB0MD0	
PCIORH	—	—	—	—	—	—	—	—	
	—	—	PC21IOR	PC20IOR	PC19IOR	PC18IOR	PC17IOR	PC16IOR	
PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR	
	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
PCCR7	—	—	—	—	—	—	—	—	
	—	—	PC25MD1	PC25MD0	—	—	PC24MD1	PC24MD0	
PCCR6	—	—	PC23MD1	PC23MD0	—	—	PC22MD1	PC22MD0	
	—	—	PC21MD1	PC21MD0	—	—	PC20MD1	PC20MD0	
PCCR5	—	—	—	PC19MD0	—	—	—	PC18MD0	
	—	—	—	PC17MD0	—	—	—	PC16MD0	
PCCR4	—	—	—	PC15MD0	—	—	—	PC14MD0	
	—	—	—	PC13MD0	—	—	PC12MD1	PC12MD0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
PCCR3	—	—	PC11MD1	PC11MD0	—	—	—	PC10MD0	PFC
	—	—	—	PC9MD0	—	—	—	PC8MD0	
PCCR2	—	—	—	PC7MD0	—	—	PC6MD1	PC6MD0	
	—	—	PC5MD1	PC5MD0	—	—	PC4MD1	PC4MD0	
PCCR1	—	—	PC3MD1	PC3MD0	—	—	PC2MD1	PC2MD0	
	—	—	—	PC1MD0	—	—	—	PC0MD0	
PDIOR	—	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
PDCR5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	PD16MD1	PD16MD0	
PDCR4	—	—	PD15MD1	PD15MD0	—	—	PD14MD1	PD14MD0	
	—	—	PD13MD1	PD13MD0	—	—	PD12MD1	PD12MD0	
PDCR3	—	—	PD11MD1	PD11MD0	—	—	PD10MD1	PD10MD0	
	—	—	PD9MD1	PD9MD0	—	—	PD8MD1	PD8MD0	
PDCR2	—	—	PD7MD1	PD7MD0	—	—	PD6MD1	PD6MD0	
	—	—	PD5MD1	PD5MD0	—	—	PD4MD1	PD4MD0	
PDCR1	—	—	—	PD3MD0	—	—	—	PD2MD0	
	—	—	—	PD1MD0	—	—	—	PD0MD0	
PECR2	—	—	—	PE7MD0	—	—	—	PE6MD0	
	—	—	—	PE5MD0	—	—	—	PE4MD0	
PECR1	—	—	—	PE3MD0	—	—	—	PE2MD0	
	—	—	—	PE1MD0	—	—	—	PE0MD0	
PFIOR	—	—	—	—	—	—	—	—	
	PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR	
PFCR2	—	—	—	PF7MD0	—	—	PF6MD1	PF6MD0	
	—	—	PF5MD1	PF5MD0	—	—	PF4MD1	PF4MD0	
PFCR1	—	—	—	PF3MD0	—	—	PF2MD1	PF2MD0	
	—	—	PF1MD1	PF1MD0	—	—	PF0MD1	PF0MD0	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	MTU2
TCR_4	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_3	—	BFE	BFB	BFA	MD3	MD2	MD1	MD0	
TMDR_4	—	BFE	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIORH_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_4	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	MTU2
TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TGCR	—	BDC	N	P	FB	WF	VF	UF	
TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	PLSP	
TOCR2	BF1	BF0	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TCNT_3									
TCNT_4									
TCDR									
TDDR									
TGRA_3									
TGRB_3									
TGRA_4									
TGRB_4									
TCNTS									
TCBR									
TGRC_3									
TGRD_3									
TGRC_4									
TGRD_4									
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
TITCR	T3AEN	3ACOR2	3ACOR1	3ACOR0	T4VEN	4VCOR2	4VCOR1	4VCOR0	MTU2
TITCNT	—	3ACNT2	3ACNT1	3ACNT0	—	4VCNT2	4VCNT1	4VCNT0	
TBTCR	—	—	—	—	—	—	BTE1	BTE0	
TDER	—	—	—	—	—	—	—	TDER	
TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4	—	—	—	—	—	—	TTSB	TTSA	
TADCR	BF1	BF0	—	—	—	—	—	—	
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE	
TADCORA_4									
TADCORB_4									
TADCOBRA_4									
TADCOBRB_4									
TWCR	CCE	—	—	—	—	—	—	WRE	
TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S	SCH4S	
TRWER	—	—	—	—	—	—	—	RWE	
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_0	—	BFE	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0									
TGRA_0									
TGRB_0									
TGRC_0									

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
TGRD_0									MTU2
TGRE_0									
TGRF_0									
TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE	
TSR2_0	—	—	—	—	—	—	TGFF	TGFE	
TBTM	—	—	—	—	—	TTSE	TTSB	TTSA	
TCR_1	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_1	—	BFE	BFB	BFA	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE	
TCR_2	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	
TMDR_2	—	BFE	BFB	BFA	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									
TCNTU_5									
TGRU_5									

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
TCRU_5	—	—	—	—	—	—	TPSC1	TPSC0	MTU2
TIORU_5	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0	
TCNTV_5									
TGRV_5									
TCRV_5	—	—	—	—	—	—	TPSC1	TPSC0	
TIORV_5	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0	
TCNTW_5									
TGRW_5									
TCRW_5	—	—	—	—	—	—	TPSC1	TPSC0	
TIORW_5	—	—	—	IOC4	IOC3	IOC2	IOC1	IOC0	
TSR_5	—	—	—	—	—	CMFU5	CMFV5	CMFW5	
TIER_5	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W	
TSTR_5	—	—	—	—	—	CSTU5	CSTV5	CSTW5	
TCNTCMPCLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W	
T8TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR
T8TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	
T8TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
T8TCSR_1	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
T8TCORA_0									
T8TCORA_1									
T8TCORB_0									
T8TCORB_1									
T8TCNT_0									
T8TCNT_1									
T8TCCR_0	—	—	—	—	TMRIS	—	ICKS1	ICKS0	
T8TCCR_1	—	—	—	—	TMRIS	—	ICKS1	ICKS0	
ADDRA									ADC
				—	—	—	—	—	
ADDRB									
				—	—	—	—	—	
ADDRC									
				—	—	—	—	—	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
ADDRD									ADC
				—	—	—	—	—	
ADDRE									
				—	—	—	—	—	
ADDRF									
				—	—	—	—	—	
ADDRG									
				—	—	—	—	—	
ADDRH									
				—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	—	TRGS1	TRGS0	—	—	
	CKS1	CKS0	MDS2	MDS1	MDS0	CH2	CH1	CH0	
DADR0									DAC
DADR1									
DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	
SCSMR_0	—	—	—	—	—	—	—	—	SCIF
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_0									
SCSCR_0	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_0									
SCFSR_0	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_0									
SCFCR_0	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_0	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_0	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_0	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_1	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_1									
SCSCR_1	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFTDR_1									SCIF
SCFSR_1	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_1									
SCFCR_1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_1	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_1	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_1	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_2	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_2									
SCSCR_2	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_2									
SCFSR_2	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_2									
SCFCR_2	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_2	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_2	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_2	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_3	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_3									
SCSCR_3	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_3									
SCFSR_3	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFRDR_3									SCIF
SCFCR_3	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_3	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_3	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_4	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_4									
SCSCR_4	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_4									
SCFSR_4	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_4									
SCFCR_4	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_4	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_4	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_5	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_5									
SCSCR_5	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_5									
SCFSR_5	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_5									
SCFCR_5	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFDR_5	—	—	—	T4	T3	T2	T1	T0	SCIF
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_5	—	—	—	—	—	—	—	—	
	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	RTSDT	
SCLSR_5									
								ORER	
SCSMR_6	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_6									
SCSCR_6	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_6									
SCFSR_6	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_6									
SCFCR_6	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_6	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	
SCSPTR_6	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_6	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SCSMR_7	—	—	—	—	—	—	—	—	
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	—	CKS1	CKS0	
SCBRR_7									
SCSCR_7	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE1	CKE0	
SCFTDR_7									
SCFSR_7	PER3	PER2	PER1	PER0	FER3	FER2	FER1	FER0	
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	
SCFRDR_7									
SCFCR_7	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFDR_7	—	—	—	T4	T3	T2	T1	T0	
	—	—	—	R4	R3	R2	R1	R0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCSPTR_7	—	—	—	—	—	—	—	—	SCIF
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT	
SCLSR_7	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
SSICR_0	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	SSI
	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0	
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	
	—	CKDV2	CKDV1	CKDV0	MUEN	—	TRMD	EN	
SSISR_0	—	—	—	DMRQ	UIRQ	OIEN	IIRQ	DIRQ	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CHNO1	CHNO0	SWNO	IDST	
SSITDR_0									
SSIRDR_0									
SSICR_1	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN	
	CHNL1	CHNL0	DWL2	DWL1	DWL0	SWL2	SWL1	SWL0	
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	
	—	CKDV2	CKDV1	CKDV0	MUEN	—	TRMD	EN	
SSISR_1	—	—	—	DMRQ	UIRQ	OIEN	IIRQ	DIRQ	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	CHNO1	CHNO0	SWNO	IDST	
SSITDR_1									
SSIRDR_1									

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
ICCR1_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC3
ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—	
ICMR_0	MLS	WAIT	—	—	BCWP	BS2	BC1	BC0	
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL_OVE	AAS	ADZ	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_0									
ICDRR_0									
NF2CYC_0	—	—	—	—	—	—	—	NF2CYC	
ICCR1_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	
ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—	
ICMR_1	MLS	WAIT	—	—	BCWP	BS2	BC1	BC0	
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL_OVE	AAS	ADZ	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_1									
ICDRR_1									
NF2CYC_1	—	—	—	—	—	—	—	NF2CYC	
ICCR1_2	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	
ICCR2_2	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—	
ICMR_2	MLS	WAIT	—	—	BCWP	BS2	BC1	BC0	
ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL_OVE	AAS	ADZ	
SAR_2	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS	
ICDRT_2									
ICDRR_2									
NF2CYC_2	—	—	—	—	—	—	—	NF2CYC	
MCR_0	MCR15	MCR14	—	—	—	TST2	TST1	TST0	RCAN-ET
	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0	
GSR_0	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
BCR1_0	TSG1_3	TSG1_2	TSG1_1	TSG1_0	—	TSG2_2	TSG2_1	TSG2_0	
	—	—	SJW1	SJW0	—	—	—	BSP	
BCR0_0	—	—	—	—	—	—	—	—	
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
IRR_0	—	—	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
IMR_0	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	RCAN-ET
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
TEC_0/REC_0	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TXPR1_0	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	
	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0	
TXPR0_0	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	
	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—	
TXCR0_0	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	
	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—	
TXACK0_0	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8	
	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—	
ABACK0_0	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8	
	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—	
RXPR0_0	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8	
	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0	
RFPR0_0	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8	
	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0	
MBIMR0_0	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8	
	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0	
UMSR0_0	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8	
	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0	
MB[0]. CONTROL0H (MCR15 = 1)	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB[0]. CONTROL0H (MCR15 = 0)	—	STDID10	STDID9	STDID8	STDID7	STDID6	STDID5	STDID4	
	STDID3	STDID2	STDID1	STDID0	RTR	IDE	EXTID17	EXTID16	
MB[0]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID:0	
MB[0]. LAFMH (MCR15 = 1)	IDE_LAFM	—	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	
	STDID_ LAFM5	STDID_ LAFM4	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	EXTID_ LAFM17	EXTID_ LAFM16	
MB[0]. LAFMH (MCR15 = 0)	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	STDID_ LAFM5	STDID_ LAFM4	
	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	—	IDE_ LAFM	EXTID_ LAFM17	EXTID_ LAFM16	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
MB[0]. LAFML	EXTID_ LAFM15	EXTID_ LAFM14	EXTID_ LAFM13	EXTID_ LAFM12	EXTID_ LAFM11	EXTID_ LAFM10	EXTID_ LAFM9	EXTID_ LAFM8	RCAN-ET
	EXTID_ LAFM7	EXTID_ LAFM6	EXTID_ LAFM5	EXTID_ LAFM4	EXTID_ LAFM3	EXTID_ LAFM2	EXTID_ LAFM1	EXTID_ LAFM0	
MB[0]. MSG_DATA[0]	MSG_DATA_0								
MB[0]. MSG_DATA[1]	MSG_DATA_1								
MB[0]. MSG_DATA[2]	MSG_DATA_2								
MB[0]. MSG_DATA[3]	MSG_DATA_3								
MB[0]. MSG_DATA[4]	MSG_DATA_4								
MB[0]. MSG_DATA[5]	MSG_DATA_5								
MB[0]. MSG_DATA[6]	MSG_DATA_6								
MB[0]. MSG_DATA[7]	MSG_DATA_7								
MB[0]. CONTROL1H	—	—	NMC	—	—	MBC2	MBC1	MBC0	
MB[0]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB[1 to 15] CONTROL0H (MCR15 = 1)	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB[1 to 15]. CONTROL0H (MCR15 = 0)	—	STDID10	STDID9	STDID8	STDID7	STDID6	STDID5	STDID4	
	STDID3	STDID2	STDID1	STDID0	RTR	IDE	EXTID17	EXTID16	
MB[1 to 15]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID:0	
MB[1 to 15]. LAFMH (MCR15 = 1)	IDE_LAFM	—	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	
	STDID_ LAFM5	STDID_ LAFM4	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	EXTID_ LAFM17	EXTID_ LAFM16	
MB[1 to 15]. LAFMH (MCR15 = 0)	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	STDID_ LAFM5	STDID_ LAFM4	
	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	—	IDE_ LAFM	EXTID_ LAFM17	EXTID_ LAFM16	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
MB[1 to 15]. LAFML	EXTID_ LAFM15	EXTID_ LAFM14	EXTID_ LAFM13	EXTID_ LAFM12	EXTID_ LAFM11	EXTID_ LAFM10	EXTID_ LAFM9	EXTID_ LAFM8	RCAN-ET
	EXTID_ LAFM7	EXTID_ LAFM6	EXTID_ LAFM5	EXTID_ LAFM4	EXTID_ LAFM3	EXTID_ LAFM2	EXTID_ LAFM1	EXTID_ LAFM0	
MB[1 to 15]. MSG_DATA[0]	MSG_DATA_0								
MB[1 to 15]. MSG_DATA[1]	MSG_DATA_1								
MB[1 to 15]. MSG_DATA[2]	MSG_DATA_2								
MB[1 to 15]. MSG_DATA[3]	MSG_DATA_3								
MB[1 to 15]. MSG_DATA[4]	MSG_DATA_4								
MB[1 to 15]. MSG_DATA[5]	MSG_DATA_5								
MB[1 to 15]. MSG_DATA[6]	MSG_DATA_6								
MB[1 to 15]. MSG_DATA[7]	MSG_DATA_7								
MB[1 to 15]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB[1 to 15]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MCR_1	MCR15	MCR14	—	—	—	TST2	TST1	TST0	
	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0	
GSR_1	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
BCR1_1	TSG1_3	TSG1_2	TSG1_1	TSG1_0	—	TSG2_2	TSG2_1	TSG2_0	
	—	—	SJW1	SJW0	—	—	—	BSP	
BCR0_1	—	—	—	—	—	—	—	—	
	BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
IRR_1	—	—	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
IMR_1	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
TEC_1/REC_1	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TXPR1_1	TXPR1_15	TXPR1_14	TXPR1_13	TXPR1_12	TXPR1_11	TXPR1_10	TXPR1_9	TXPR1_8	
	TXPR1_7	TXPR1_6	TXPR1_5	TXPR1_4	TXPR1_3	TXPR1_2	TXPR1_1	TXPR1_0	

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
TXPR0_1	TXPR0_15	TXPR0_14	TXPR0_13	TXPR0_12	TXPR0_11	TXPR0_10	TXPR0_9	TXPR0_8	RCAN-ET
	TXPR0_7	TXPR0_6	TXPR0_5	TXPR0_4	TXPR0_3	TXPR0_2	TXPR0_1	—	
TXCR0_1	TXCR0_15	TXCR0_14	TXCR0_13	TXCR0_12	TXCR0_11	TXCR0_10	TXCR0_9	TXCR0_8	
	TXCR0_7	TXCR0_6	TXCR0_5	TXCR0_4	TXCR0_3	TXCR0_2	TXCR0_1	—	
TXACK0_1	TXACK0_15	TXACK0_14	TXACK0_13	TXACK0_12	TXACK0_11	TXACK0_10	TXACK0_9	TXACK0_8	
	TXACK0_7	TXACK0_6	TXACK0_5	TXACK0_4	TXACK0_3	TXACK0_2	TXACK0_1	—	
ABACK0_1	ABACK0_15	ABACK0_14	ABACK0_13	ABACK0_12	ABACK0_11	ABACK0_10	ABACK0_9	ABACK0_8	
	ABACK0_7	ABACK0_6	ABACK0_5	ABACK0_4	ABACK0_3	ABACK0_2	ABACK0_1	—	
RXPR0_1	RXPR0_15	RXPR0_14	RXPR0_13	RXPR0_12	RXPR0_11	RXPR0_10	RXPR0_9	RXPR0_8	
	RXPR0_7	RXPR0_6	RXPR0_5	RXPR0_4	RXPR0_3	RXPR0_2	RXPR0_1	RXPR0_0	
RFPR0_1	RFPR0_15	RFPR0_14	RFPR0_13	RFPR0_12	RFPR0_11	RFPR0_10	RFPR0_9	RFPR0_8	
	RFPR0_7	RFPR0_6	RFPR0_5	RFPR0_4	RFPR0_3	RFPR0_2	RFPR0_1	RFPR0_0	
MBIMR0_1	MBIMR0_15	MBIMR0_14	MBIMR0_13	MBIMR0_12	MBIMR0_11	MBIMR0_10	MBIMR0_9	MBIMR0_8	
	MBIMR0_7	MBIMR0_6	MBIMR0_5	MBIMR0_4	MBIMR0_3	MBIMR0_2	MBIMR0_1	MBIMR0_0	
UMSR0_1	UMSR0_15	UMSR0_14	UMSR0_13	UMSR0_12	UMSR0_11	UMSR0_10	UMSR0_9	UMSR0_8	
	UMSR0_7	UMSR0_6	UMSR0_5	UMSR0_4	UMSR0_3	UMSR0_2	UMSR0_1	UMSR0_0	
MB[0]. CONTROL0H (MCR15 = 1)	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB[0]. CONTROL0H (MCR15 = 0)	—	STDID10	STDID9	STDID8	STDID7	STDID6	STDID5	STDID4	
	STDID3	STDID2	STDID1	STDID0	RTR	IDE	EXTID17	EXTID16	
MB[0]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID:0	
MB[0]. LAFMH (MCR15 = 1)	IDE_LAFM	—	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	
	STDID_ LAFM5	STDID_ LAFM4	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	EXTID_ LAFM17	EXTID_ LAFM16	
MB[0]. LAFMH (MCR15 = 0)	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	STDID_ LAFM5	STDID_ LAFM4	
	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	—	IDE_ LAFM	EXTID_ LAFM17	EXTID_ LAFM16	
MB[0]. LAFML	EXTID_ LAFM15	EXTID_ LAFM14	EXTID_ LAFM13	EXTID_ LAFM12	EXTID_ LAFM11	EXTID_ LAFM10	EXTID_ LAFM9	EXTID_ LAFM8	
	EXTID_ LAFM7	EXTID_ LAFM6	EXTID_ LAFM5	EXTID_ LAFM4	EXTID_ LAFM3	EXTID_ LAFM2	EXTID_ LAFM1	EXTID_ LAFM0	
MB[0]. MSG_DATA[0]	MSG_DATA_0								
MB[0]. MSG_DATA[1]	MSG_DATA_1								

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
MB[0]. MSG_DATA[2]	MSG_DATA_2								RCAN-ET
MB[0]. MSG_DATA[3]	MSG_DATA_3								
MB[0]. MSG_DATA[4]	MSG_DATA_4								
MB[0]. MSG_DATA[5]	MSG_DATA_5								
MB[0]. MSG_DATA[6]	MSG_DATA_6								
MB[0]. MSG_DATA[7]	MSG_DATA_7								
MB[0]. CONTROL1H	—	—	NMC	—	—	MBC2	MBC1	MBC0	
MB[0]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
MB[1 to 15]. CONTROL0H (MCR15 = 1)	IDE	RTR	—	STDID10	STDID9	STDID8	STDID7	STDID6	
	STDID5	STDID4	STDID3	STDID2	STDID1	STDID0	EXTID17	EXTID16	
MB[1 to 15]. CONTROL0H (MCR15 = 0)	—	STDID10	STDID9	STDID8	STDID7	STDID6	STDID5	STDID4	
	STDID3	STDID2	STDID1	STDID0	RTR	IDE	EXTID17	EXTID16	
MB[1 to 15]. CONTROL0L	EXTID15	EXTID14	EXTID13	EXTID12	EXTID11	EXTID10	EXTID9	EXTID8	
	EXTID7	EXTID6	EXTID5	EXTID4	EXTID3	EXTID2	EXTID1	EXTID:0	
MB[1 to 15]. LAFMH (MCR15 = 1)	IDE_LAFM	—	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	
	STDID_ LAFM5	STDID_ LAFM4	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	EXTID_ LAFM17	EXTID_ LAFM16	
MB[1 to 15]. LAFMH (MCR15 = 0)	—	STDID_ LAFM10	STDID_ LAFM9	STDID_ LAFM8	STDID_ LAFM7	STDID_ LAFM6	STDID_ LAFM5	STDID_ LAFM4	
	STDID_ LAFM3	STDID_ LAFM2	STDID_ LAFM1	STDID_ LAFM0	—	IDE_ LAFM	EXTID_ LAFM17	EXTID_ LAFM16	
MB[1 to 15]. LAFML	EXTID_ LAFM15	EXTID_ LAFM14	EXTID_ LAFM13	EXTID_ LAFM12	EXTID_ LAFM11	EXTID_ LAFM10	EXTID_ LAFM9	EXTID_ LAFM8	
	EXTID_ LAFM7	EXTID_ LAFM6	EXTID_ LAFM5	EXTID_ LAFM4	EXTID_ LAFM3	EXTID_ LAFM2	EXTID_ LAFM1	EXTID_ LAFM0	
MB[1 to 15]. MSG_DATA[0]	MSG_DATA_0								
MB[1 to 15]. MSG_DATA[1]	MSG_DATA_1								
MB[1 to 15]. MSG_DATA[2]	MSG_DATA_2								

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
MB[1 to 15]. MSG_DATA[3]					MSG_DATA_3				RCAN-ET
MB[1 to 15]. MSG_DATA[4]					MSG_DATA_4				
MB[1 to 15]. MSG_DATA[5]					MSG_DATA_5				
MB[1 to 15]. MSG_DATA[6]					MSG_DATA_6				
MB[1 to 15]. MSG_DATA[7]					MSG_DATA_7				
MB[1 to 15]. CONTROL1H	—	—	NMC	ATX	DART	MBC2	MBC1	MBC0	
MB[1 to 15]. CONTROL1L	—	—	—	—	DLC3	DLC2	DLC1	DLC0	
DREQER0	—	—	IIC2TX	IIC2RX	IIC1TX	IIC1RX	IIC0TX	IIC0RX	INTC
DREQER1	SCIF3TX	SCIF3RX	SCIF2TX	SCIF2RX	SCIF1TX	SCIF1RX	SCIF0TX	SCIF0RX	
DREQER2	SCIF7TX	SCIF7RX	SCIF6TX	SCIF6RX	SCIF5TX	SCIF5RX	SCIF4TX	SCIF4RX	
DREQER3	ADC	MTU4	MTU3	MTU2	MTU1	MTU0	RCAN1	RCAN0	
DSFR	IOKEEP	—	—	—	—	—	MRESF	NMIF	SYSTEM
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
DSCNT	—	—	—	—	—	CKS2	CKS1	CKS0	
RAMKP	—	—	—	—	RAMKP3	RAMKP2	RAMKP1	RAMKP0	

28.3 Register States in Each Operating Mode

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
SYCBEEN	Initialized	Retained	Retained	Initialized*[1]	—	Retained	Bus Monitor
SYCBESTS1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
SYCBESTS2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
SYCBESW	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS0CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	BSC
CS0REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS1CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS1REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS2CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS2REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS3CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS3REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS4CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS4REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS5CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS5REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS6CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS6REC	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
SDC0CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
SDC1CNT	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CSMOD0	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS1WCNT0	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS2WCNT0	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CSMOD1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS1WCNT1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS2WCNT1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CSMOD2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS1WCNT2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS2WCNT2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CSMOD3	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS1WCNT3	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CS2WCNT3	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
CSMOD4	Initialized	Retained	Retained	Initialized*[1]	—	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
CS1WCNT4	Initialized	Retained	Retained	Initialized*1	—	Retained	BSC
CS2WCNT4	Initialized	Retained	Retained	Initialized*1	—	Retained	
CSMOD5	Initialized	Retained	Retained	Initialized*1	—	Retained	
CS1WCNT5	Initialized	Retained	Retained	Initialized*1	—	Retained	
CS2WCNT5	Initialized	Retained	Retained	Initialized*1	—	Retained	
CSMOD6	Initialized	Retained	Retained	Initialized*1	—	Retained	
CS1WCNT6	Initialized	Retained	Retained	Initialized*1	—	Retained	
CS2WCNT6	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDRFCNT0	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDRFCNT1	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDIR0	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDIR1	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDPWDCNT	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDDPWDCNT	Initialized	Retained	Retained	Initialized*1	—	Retained	
SD0ADR	Initialized	Retained	Retained	Initialized*1	—	Retained	
SD0TR	Initialized	Retained	Retained	Initialized*1	—	Retained	
SD0MOD	Initialized	Retained	Retained	Initialized*1	—	Retained	
SD1ADR	Initialized	Retained	Retained	Initialized*1	—	Retained	
SD1TR	Initialized	Retained	Retained	Initialized*1	—	Retained	
SD1MOD	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDSTR	Initialized	Retained	Retained	Initialized*1	—	Retained	
SDCKSCNT	Initialized	Retained	Retained	Initialized*1	—	Retained	
DMCSADR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	DMAC
DMCDADR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
DMCBCT3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	DMAC
DMMOD3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCSADR7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCDADR7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCBCT7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMMOD7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
DMRSADR6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	DMAC
DMRDADR6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRSADR7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRDADR7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMRBCT7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB5	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB6	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTA7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMCNTB7	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMSCNT	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMICNT	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMICNTA	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMISTS	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMEDET	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
DMASTS	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BAR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	UBC
BAMR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BDR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BDMR0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BAR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BAMR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BDR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
BDMR1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
BBR0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	UBC
BBR1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
BRCR	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
CCR1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	Cache
CCR2	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
ACSWR	Initialized	Retained	Retained	Initialized*[1]	—	Retained	BSC
SDIR*[2]	Initialized	Retained	Retained	Initialized*[1]	—	Retained	H-UDI
ICR0	Initialized	Retained	Retained	Initialized*[1]	—	Retained	INTC
ICR1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
ICR2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IRQRR	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PINTER	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PIRR	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IBCR	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IBNR	Initialized	Retained*[3]	Retained	Initialized*[1]	—	Retained	
IPR01	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR02	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR05	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR06	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR07	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR08	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR09	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR10	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR11	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR12	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR13	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR14	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR15	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
IPR16	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
WTCSR	Initialized	Retained	Initialized	Initialized*[1]	—	Retained	WDT
WTCNT	Initialized	Retained	Initialized	Initialized*[1]	—	Retained	
WRCSR	Initialized*[4]	Retained	Initialized	Initialized*[1]	—	Retained	
FRQCR	Initialized*[4]	Retained	Retained	Initialized*[1]	—	Retained	CPG
STBCR	Initialized	Retained	Retained	Initialized*[1]	—	Retained	SYSTEM
STBCR2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
SYSCR1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
SYSCR2	Initialized	Retained	Retained	Initialized*1	—	Retained	SYSTEM
STBCR3	Initialized	Retained	Retained	Initialized*1	—	Retained	
STBCR4	Initialized	Retained	Retained	Initialized*1	—	Retained	
STBCR5	Initialized	Retained	Retained	Initialized*1	—	Retained	
R64CNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	RTC
RSECCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RMINCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RHRCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RWKCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RDAYCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RMONCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RYRCNT	Retained*5	Retained*5	Retained*5	Retained*5	Retained	Retained*5	
RSECAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RMINAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RHRAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RWKAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RDAYAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RMONAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RCR1	Initialized	Initialized	Retained	Initialized	Retained	Retained	
RCR2	Initialized	Initialized*6	Retained	Initialized	Retained	Retained	
RYRAR	Retained	Retained	Retained	Retained	Retained	Retained	
RCR3	Initialized	Retained	Retained	Initialized	Retained	Retained	
PADRH	Initialized	Retained	Retained	Initialized*1	—	Retained	I/O ports
PADRL	Initialized	Retained	Retained	Initialized*1	—	Retained	
PAPRH	Undefined	Retained	Retained	Initialized*1	—	Retained	
PAPRL	Undefined	Retained	Retained	Initialized*1	—	Retained	
PBDRH	Initialized	Retained	Retained	Initialized*1	—	Retained	
PBDRL	Initialized	Retained	Retained	Initialized*1	—	Retained	
PBPRH	Undefined	Retained	Retained	Initialized*1	—	Retained	
PBPRL	Undefined	Retained	Retained	Initialized*1	—	Retained	
PCDRH	Initialized	Retained	Retained	Initialized*1	—	Retained	
PCDRL	Initialized	Retained	Retained	Initialized*1	—	Retained	
PCPRH	Undefined	Retained	Retained	Initialized*1	—	Retained	
PCPRL	Undefined	Retained	Retained	Initialized*1	—	Retained	
PDDRH	Initialized	Retained	Retained	Initialized*1	—	Retained	
PDDRL	Initialized	Retained	Retained	Initialized*1	—	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
PDPRH	Undefined	Retained	Retained	Initialized*[1]	—	Retained	I/O ports
PDPRL	Undefined	Retained	Retained	Initialized*[1]	—	Retained	
PEPRL	Undefined	Retained	Retained	Initialized*[1]	—	Retained	
PFDR	Undefined	Retained	Retained	Initialized*[1]	—	Retained	
PFPR	Undefined	Retained	Retained	Initialized*[1]	—	Retained	
PAIORH	Initialized	Retained	Retained	Initialized*[1]	—	Retained	PFC
PAIORL	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR8	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR7	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR6	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR5	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR4	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR3	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PACR1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBIORH	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBIORL	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR8	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR7	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR6	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR5	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR4	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR3	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PBCR1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCIORH	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCIORL	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR7	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR6	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR5	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR4	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR3	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR2	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PCCR1	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PDIORH	Initialized	Retained	Retained	Initialized*[1]	—	Retained	
PDIORL	Initialized	Retained	Retained	Initialized*[1]	—	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
PDCR5	Initialized	Retained	Retained	Initialized*1	—	Retained	PFC
PDCR4	Initialized	Retained	Retained	Initialized*1	—	Retained	
PDCR3	Initialized	Retained	Retained	Initialized*1	—	Retained	
PDCR2	Initialized	Retained	Retained	Initialized*1	—	Retained	
PDCR1	Initialized	Retained	Retained	Initialized*1	—	Retained	
PECR2	Initialized	Retained	Retained	Initialized*1	—	Retained	
PECR1	Initialized	Retained	Retained	Initialized*1	—	Retained	
PFIOR	Initialized	Retained	Retained	Initialized*1	—	Retained	
PFCR2	Initialized	Retained	Retained	Initialized*1	—	Retained	
PFCR1	Initialized	Retained	Retained	Initialized*1	—	Retained	
TCR_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	MTU2
TCR_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TMDR_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TMDR_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORH_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORL_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORH_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORL_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIER_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIER_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TOER	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGCR	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TOCR1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TOCR2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNT_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNT_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCDR	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TDDR	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRA_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRB_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRA_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRB_4	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNTS	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCBR	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRC_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRD_3	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
TGRC_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	MTU2
TGRD_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TSR_3	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TSR_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TITCR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TITCNT	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TBTCR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TDER	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TOLBR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TBTM_3	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TBTM_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TADCR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TADCORA_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TADCORB_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TADCOBRA_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TADCOBRB_4	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TSYCR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TWCR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TSTR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TSYR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TCSYSTR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TRWER	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TCR_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TMDR_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TIORH_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TIORL_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TIER_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TSR_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TCNT_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TGRA_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TGRB_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TGRC_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TGRD_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TGRE_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TGRF_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
TIER2_0	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
TSR2_0	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	MTU2
TBTM	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCR_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TMDR_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIOR_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIER_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TSR_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNT_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRA_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRB_1	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TICCR	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCR_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TMDR_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIOR_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIER_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TSR_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNT_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRA_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRB_2	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNTU_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRU_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCRU_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORU_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNTV_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRV_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCRV_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORV_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNTW_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TGRW_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCRW_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIORW_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TSR_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TIER_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TSTR_5	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	
TCNTCMPCLR	Initialized	Retained	Initialized	Initialized*1	Initialized	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
T8TCR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	TMR
T8TCR_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCSR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCSR_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCORA_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCORA_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCORB_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCORB_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCNT_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCNT_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCCR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
T8TCCR_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
ADDRA	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	ADC
ADDRB	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADDRC	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADDRD	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADDRE	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADDRF	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADDRG	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADDRH	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
ADCSR	Initialized	Retained	Initialized	Initialized*[1]	Initialized	Retained	
DADR0	Initialized	Retained	Retained	Initialized*[1]	Initialized	Retained	DAC
DADR1	Initialized	Retained	Retained	Initialized*[1]	Initialized	Retained	
DACR	Initialized	Retained	Retained	Initialized*[1]	Initialized	Retained	
SCSMR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	SCIF
SCBRR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSCR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFTDR_0	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFSR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFRDR_0	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFCR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFDR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSPTR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCLSR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSMR_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCBRR_1	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
SCSCR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	SCIF
SCFTDR_1	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFSR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFRDR_1	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFCR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFDR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSPTR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCLSR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSMR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCBRR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSCR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFTDR_2	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFSR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFRDR_2	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFCR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFDR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSPTR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCLSR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSMR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCBRR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSCR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFTDR_3	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFSR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFRDR_3	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFCR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFDR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSPTR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCLSR_3	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSMR_4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCBRR_4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCSCR_4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFTDR_4	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFSR_4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFRDR_4	Undefined	Retained	Retained	Initialized*1	Retained	Retained	
SCFCR_4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SCFDR_4	Initialized	Retained	Retained	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
SCSPTR_4	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	SCIF
SCLSR_4	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSMR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCBRR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSCR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFTDR_5	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFSR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFRDR_5	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFCR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFDR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSPTR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCLSR_5	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSMR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCBRR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSCR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFTDR_6	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFSR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFRDR_6	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFCR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFDR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSPTR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCLSR_6	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSMR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCBRR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSCR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFTDR_7	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFSR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFRDR_7	Undefined	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFCR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCFDR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCSPTR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SCLSR_7	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SSICR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	SSI
SSISR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SSITDR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	
SSIRDR_0	Initialized	Retained	Retained	Initialized*[1]	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
SSICR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	SSI
SSISR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SSITDR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SSIRDR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICCR1_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	IIC3
ICCR2_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICMR_0	Initialized	Retained	Retained*7	Initialized*1	Retained*7	Retained	
ICIER_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICSR_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SAR_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICDRT_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICDRR_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
NF2CYC_0	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICCR1_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICCR2_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICMR_1	Initialized	Retained	Retained*7	Initialized*1	Retained*7	Retained	
ICIER_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICSR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SAR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICDRT_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICDRR_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
NF2CYC_1	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICCR1_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICCR2_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICMR_2	Initialized	Retained	Retained*7	Initialized*1	Retained*7	Retained	
ICIER_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICSR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
SAR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICDRT_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
ICDRR_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
NF2CYC_2	Initialized	Retained	Retained	Initialized*1	Retained	Retained	
MCR_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	RCAN-ET
GSR_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
BCR1_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
BCR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
IRR_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
IMR_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	RCAN-ET
TEC_0/REC_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXPR1_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXPR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXCR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXACK0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
ABACK0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
RXPR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
RFPR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MBIMR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
UMSR0_0	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL0H	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL0L	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. LAFMH	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. LAFML	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[0]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[1]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[2]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[3]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[4]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[5]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[6]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[7]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL1H	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL1L	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
MCR_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	RCAN-ET
GSR_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
BCR1_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
BCR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
IRR_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
IMR_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TEC_1/REC_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXPR1_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXPR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXCR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
TXACK0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
ABACK0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
RXPR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
RFPR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MBIMR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
UMSR0_1	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL0H	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL0L	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. LAFMH	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. LAFML	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[0]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[1]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[2]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[3]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[4]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[5]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. MSG_DATA[6]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
MB[0 to 15]. MSG_DATA[7]	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	RCAN-ET
MB[0 to 15]. CONTROL1H	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
MB[0 to 15]. CONTROL1L	Initialized	Retained	Initialized	Initialized*1	Retained	Retained	
DREQER0	Initialized	Retained	Retained	Initialized*1	—	Retained	INTC
DREQER1	Initialized	Retained	Retained	Initialized*1	—	Retained	
DREQER2	Initialized	Retained	Retained	Initialized*1	—	Retained	
DREQER3	Initialized	Retained	Retained	Initialized*1	—	Retained	
DSFR	Initialized	Retained	Retained	Retained	—	Retained	SYSTEM
DSCNT	Initialized	Retained	Retained	Initialized*1	—	Retained	
RAMKP	Initialized	Retained	Retained	Initialized*1	—	Retained	

Notes: 1. Not initialized in deep standby mode. But initialized after deep standby mode is released because a power-on reset exception handling is executed.
2. Initialized by $\overline{\text{UDTRST}}$ assertion or in the Test-Logic-Reset state of the TAP controller.
3. Bits BN[3:0] are initialized.
4. Retains the previous value after an internal power-on reset by means of the WDT.
5. Counting up continues.
6. Bits RTCEN and START are retained.
7. Bits BC[3:0] are initialized.
8. The ENB bit is initialized.

Section 29 Electrical Characteristics

29.1 Absolute Maximum Ratings

Table 29.1 lists the absolute maximum ratings.

Table 29.1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage (I/O)		PV_{CC}	–0.3 to 4.6	V
Power supply voltage (Internal)		$V_{CC}R$		
Power supply voltage (PLL)		$PLLV_{CC}$		
Analog power supply voltage		AV_{CC}	–0.3 to 4.6	V
Analog reference voltage		AV_{ref}	–0.3 to AV_{CC} +0.3	V
Input voltage	Analog input pin	V_{AN}	–0.3 to AV_{CC} +0.3	V
	PC22 to PC25, PD15, PD16	V_{in}	–0.3 to 5.5	V
	Other pins	V_{in}	–0.3 to PV_{CC} +0.3	V
Operating temperature		T_{opr}	–20 to +70 (Regular specifications)	°C
			–20 to +85 (Wide-range specifications)	
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

29.2 DC Characteristics

Tables 29.2 and 29.3 list DC characteristics.

Table 29.2 DC Characteristics (1) [Common Items] [Regular Specifications]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V ≤ $AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*	Normal operation	I_{CC}	—	120	180	mA	Iϕ = 120 MHz
			—	100	160	mA	Iϕ = 100 MHz
			—	80	140	mA	Iϕ = 80 MHz
	Sleep mode	I_{sleep}	—	60	140	mA	Iϕ = 120 MHz
			—	50	130	mA	Iϕ = 100 MHz
			—	45	125	mA	Iϕ = 80 MHz
	Software standby mode	I_{stby}	—	5	30	mA	$T_a > 50°C$
			—	1.5	20	mA	$T_a \leq 50°C$
	Deep standby mode	I_{dstby}	—	80	100	μA	$T_a > 50°C$ RAM: 0 Kbyte retained
			—	300	750	μA	$T_a > 50°C$ RAM: 8 Kbytes retained
			—	500	1500	μA	$T_a > 50°C$ RAM: 16 Kbytes retained
			—	750	2250	μA	$T_a > 50°C$ RAM: 24 Kbytes retained
			—	1000	3000	μA	$T_a > 50°C$ RAM: 32 Kbytes retained
			—	50	75	μA	$T_a \leq 50°C$ RAM: 0 Kbyte retained
			—	70	300	μA	$T_a \leq 50°C$ RAM: 8 Kbytes retained
			—	80	500	μA	$T_a \leq 50°C$ RAM: 16 Kbytes retained
			—	90	750	μA	$T_a \leq 50°C$ RAM: 24 Kbytes retained
			—	100	1000	μA	$T_a \leq 50°C$ RAM: 32 Kbytes retained

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	All input pins (except PC22 to PC25, PD15, PD16, PE0 to PE7, EXTAL, AUDIO_X1, and RTC_X1)	$\|I_{in}\|$	—	—	1.0	μA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	PC22 to PC25, PD15, PD16		—	—	20	μA	
Three-state leakage current	All input/output pins, output pins (off state)	$\|IT_{SI}\|$	—	—	1.0	μA	V_{in} = 0.5 to PV_{CC} – 0.5 V
Input pull-up MOS current	$\overline{\text{UDTRST}}$, UDTMS, UDTDI, UDTCK, $\overline{\text{ASEBRK/ASEBRKAK}}$	-Ip	10	—	150	μA	V_{in} = 0 V
Input capacitance	All pins	C_{in}	—	—	20	pF	
Analog power supply current	During A/D or D/A conversion	AI_{CC}	—	1	2	mA	
	Waiting for A/D or D/A conversion		—	1	2	μA	
Analog reference voltage current		AI_{ref}	—	2	3	mA	

Caution: When the A/D converter or D/A converter is not in use, the AV_{CC} and AV_{SS} pins should not be open.

Note: * Supply current values are values when all of the output pins and pins with the pull-up function ($\overline{\text{UDTRST}}$, UDTMS, UDTDI, UDTCK, $\overline{\text{ASEBRK/ASEBRKAK}}$) are unloaded and represent the total current supplied to the PV_{CC}, $V_{CC}R$, and $PLLV_{CC}$ systems. Reference values are given under “Typ.”

Table 29.2 DC Characteristics (2) [Common Items] [Wide-Range Specifications]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current*	Normal operation	I_{CC}	—	100	160	mA	Iϕ = 100 MHz
			—	80	140	mA	Iϕ = 80 MHz
	Sleep mode	I_{sleep}	—	50	130	mA	Iϕ = 100 MHz
			—	45	125	mA	Iϕ = 80 MHz
	Software standby mode	I_{stby}	—	5	40	mA	$T_a > 50$°C
			—	1.5	20	mA	$T_a \leq 50$°C
	Deep standby mode	I_{dstby}	—	80	100	μA	$T_a > 50$°C RAM: 0 Kbyte retained
			—	300	1000	μA	$T_a > 50$°C RAM: 8 Kbytes retained
			—	500	2000	μA	$T_a > 50$°C RAM: 16 Kbytes retained
			—	750	3000	μA	$T_a > 50$°C RAM: 24 Kbytes retained
			—	1000	4000	μA	$T_a > 50$°C RAM: 32 Kbytes retained
			—	50	75	μA	$T_a \leq 50$°C RAM: 0 Kbyte retained
			—	70	300	μA	$T_a \leq 50$°C RAM: 8 Kbytes retained
			—	80	500	μA	$T_a \leq 50$°C RAM: 16 Kbytes retained
			—	90	750	μA	$T_a \leq 50$°C RAM: 24 Kbytes retained
			—	100	1000	μA	$T_a \leq 50$°C RAM: 32 Kbytes retained

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	All input pins (except PC22 to PC25, PD15, PD16, PE0 to PE7, EXTAL, AUDIO_X1, and RTC_X1)	$\|I_{in}\|$	—	—	1.0	μA	V_{in} = 0.5 to PV_{CC} – 0.5 V
	PC22 to PC25, PD15, PD16		—	—	20	μA	
Three-state leakage current	All input/output pins, output pins (off state)	$\|IT_{SI}\|$	—	—	1.0	μA	V_{in} = 0.5 to PV_{CC} – 0.5 V
Input pull-up MOS current	$\overline{\text{UDTRST}}$, UDTMS, UDTDI, UDTCK, and $\overline{\text{ASEBRK/ASEBRKAK}}$	-Ip	10	—	150	μA	V_{in} = 0 V
Input capacitance	All pins	C_{in}	—	—	20	pF	
Analog power supply current	During A/D or D/A conversion	AI_{CC}	—	1	2	mA	
	Waiting for A/D or D/A conversion		—	1	2	μA	
Analog reference voltage current		AI_{ref}	—	2	3	mA	

Caution: When the A/D converter or D/A converter is not in use, the AV_{CC} and AV_{SS} pins should not be open.

Note: * Supply current values are values when all of the output pins and pins with the pull-up function ($\overline{\text{UDTRST}}$, UDTMS, UDTDI, UDTCK, $\overline{\text{ASEBRK/ASEBRKAK}}$) are unloaded and represent the total current supplied to the PV_{CC}, $V_{CC}R$, and $PLLV_{CC}$ systems. Reference values are given under “Typ.”

Table 29.2 DC Characteristics (3) [Except for I²C-Related Pins*1]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V ≤ $AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD1, MD0, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{UDTRST}$, $\overline{ASEBRK}$/$\overline{ASEBRKAK}$, EXTAL, CKIO, AUDIO_X1, RTC_X1	V_{IH}	$PV_{CC} - 0.5$	—	$PV_{CC} + 0.3$	V	
	PF7 to PF0		2.2	—	$AV_{CC} + 0.3$		
	Input pins other than above (excluding Schmitt pins)		2.2	—	$PV_{CC} + 0.3$		
Input low voltage	$\overline{RES}$, $\overline{MRES}$, NMI, MD1, MD0, MD_CLK1, MD_CLK0, $\overline{ASEMD}$, $\overline{UDTRST}$, $\overline{ASEBRK}$/$\overline{ASEBRKAK}$, EXTAL, CKIO, AUDIO_X1, RTC_X1	V_{IL}	–0.3	—	0.5	V	
	Input pins other than above (excluding Schmitt pins)		–0.3	—	0.8		

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, SCK7 to SCK0, RxD7 to RxD0, IRQ7 to IRQ0*[2], PINT7 to PINT0	V_T^+ (V_{IH})	PV_{CC} – 0.5	—	—	V	
		V_T^- (V_{IL})	—	—	0.5	V	
		$V_T^+ - V_T^-$	0.2	—	—	V	
Output high voltage	All output pins	V_{OH}	PV_{CC} – 0.5	—	—	V	I_{OH} = –200 μA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
RAM standby voltage		V_{RAM}	3.0	—	—	V	
Power-supply start voltage		VCC_{START}	—	0	0.8	V	
Power-supply rising gradient		SVCC	—	—	20	ms/V	

Notes: 1. Pins (open-drain pins): PC22/IRQ0/SCL0/DREQ2, PC23/IRQ1/SDA0, PC24/IRQ2/SCL1, PC25/IRQ3/SDA1, PD15/SDA2, and PD16/SCL2

2. Except (PC22/)IRQ0, (PC23/)IRQ1, (PC24/)IRQ2, and (PC25/)IRQ3

Table 29.2 DC Characteristics (4) [I²C-Related Pins*]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V ≤ $AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input high voltage	PC22/DREQ2, PC23, PC24, PC25, PD15, PD16	V_{IH}	2.2	—	PV_{CC} + 0.3	V	
Input low voltage		V_{IL}	–0.3	—	0.8	V	
Schmitt trigger input characteristics	IRQ0/SCL0, IRQ1/SDA0, IRQ2/SCL1, IRQ3/SDA1, SDA2, ASCL2	V_T^+ (V_{IH})	$PV_{CC} \times 0.7$	—	5.5	V	
		V_T^- (V_{IL})	–0.3	—	$PV_{CC} \times 0.3$	V	
		$V_T^+ - V_T^-$	$PV_{CC} \times 0.05$	—	—	V	
Output low voltage	SCL0 to SCL2, SDA0 to SDA2	V_{OL}	—	—	0.4	V	I_{OL} = 3.0 mA

Note: * Pins (open-drain pins): PC22/IRQ0/SCL0/DREQ2, PC23/IRQ1/SDA0, PC24/IRQ2/SCL1, PC25/IRQ3/SDA1, PD15/SDA2, and PD16/SCL2

Table 29.3 Permissible Output Currents (1) [Common Items]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V ≤ $AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0 to SCL2, SDA0 to SDA2	I_{OL}	—	—	10	mA
	Other than above				2	
Permissible output low current (total)		ΣI_{OL}	—	—	150	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	50	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in the table above.

Table 29.3 Permissible Output Currents (2) [Wide-Range Specifications]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V, I$\phi \leq 80$ MHz

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0 to SCL2, SDA0 to SDA2	I_{OL}	—	—	10	mA
	Other than above				2	
Permissible output low current (total)		ΣI_{OL}	—	—	150	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	50	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in the table above.

Table 29.3 Permissible Output Currents (3) [Wide-Range Specifications]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V, 80 MHz < I$\phi \leq 100$ MHz

Item		Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (per pin)	SCL0 to SCL2, SDA0 to SDA2	I_{OL}	—	—	10	mA
	Other than above				2	
Permissible output low current (total)		ΣI_{OL}	—	—	50	mA
Permissible output high current (per pin)		$-I_{OH}$	—	—	2	mA
Permissible output high current (total)		$\Sigma -I_{OH}$	—	—	50	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in the table above.

29.3 AC Characteristics

Signals input to this LSI are basically handled as signals in synchronization with a clock. The setup and hold times for input pins must be followed.

Table 29.4 Maximum Operating Frequency

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	20	—	120	MHz	Regular specifications
					100		Wide-range specifications
	Bus clock (Bϕ)		20	—	60		
	Peripheral clock (Pϕ)		5	—	40		

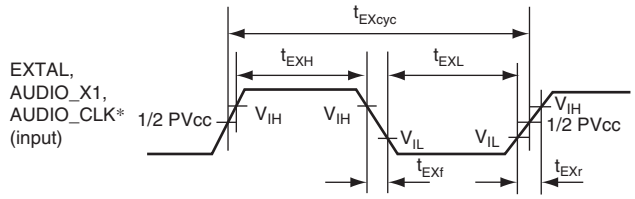
29.3.1 Clock Timing

Table 29.5 Clock Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL, XTAL crystal oscillator frequency (clock mode 0)	—	10	15	MHz	
EXTAL, XTAL crystal oscillator frequency (clock mode 2)	—	10	20	MHz	
AUDIO_X1, AUDIO_X2 crystal oscillator frequency	—	10	25	MHz	
EXTAL clock input frequency (clock mode 0)	f_{EX}	10	15	MHz	Figure 29.1
EXTAL clock input frequency (clock mode 2)	f_{EX}	10	30	MHz	

Item	Symbol	Min.	Max.	Unit	Figure
EXTAL clock input cycle time (clock mode 0)	f_{EXcyc}	66.67	100	ns	Figure 29.1
EXTAL clock input cycle time (clock mode 2)	f_{EXcyc}	33.33	100	ns	
AUDIO_X1, AUDIO_CLK clock input frequency	f_{EX}	1	40	MHz	
AUDIO_X1, AUDIO_CLK clock input cycle time	t_{EXcyc}	25	1000	ns	
EXTAL, AUDIO_X1, AUDIO_CLK clock input pulse low width	t_{EXL}	0.4	0.6	t_{cyc}	
EXTAL, AUDIO_X1, AUDIO_CLK clock input pulse high width	t_{EXH}	0.4	0.6	t_{cyc}	
EXTAL, AUDIO_X1, AUDIO_CLK clock input rise time	t_{EXr}	—	4	ns	
EXTAL, AUDIO_X1, AUDIO_CLK clock input fall time	t_{EXf}	—	4	ns	
CKIO clock input frequency	f_{CK}	20	60	MHz	Figure 29.2
CKIO clock input cycle time	t_{CKIcyc}	16.67	50	ns	
CKIO clock input pulse low width	t_{CKIL}	0.4	0.6	t_{CKIcyc}	
CKIO clock input pulse high width	t_{CKIH}	0.4	0.6	t_{CKIcyc}	
CKIO clock input rise time	t_{CKIr}	—	3	ns	
CKIO clock input fall time	t_{CKIf}	—	3	ns	
CKIO clock output frequency	f_{OP}	20	60	MHz	Figure 29.3
CKIO clock output cycle time	t_{cyc}	16.67	50	ns	
CKIO clock output pulse low width	t_{CKOL}	$t_{cyc}/2 - t_{CKOr}$	—	ns	
CKIO clock output pulse high width	t_{CKOH}	$t_{cyc}/2 - t_{CKOf}$	—	ns	
CKIO clock output rise time	t_{CKOr}	—	3	ns	
CKIO clock output fall time	t_{CKOf}	—	3	ns	
Power-on oscillation settling time	t_{OSC1}	10	—	ms	Figure 29.4
Oscillation settling time on return from standby 1	t_{OSC2}	10	—	ms	Figure 29.5
Oscillation settling time on return from standby 2	t_{OSC3}	10	—	ms	Figure 29.6
RTC clock oscillation settling time	t_{ROSC}	3	—	s	Figure 29.7



Note: * When the clock is input on the EXTAL, AUDIO_X1, or AUDIO_CLK pin.

Figure 29.1 EXTAL, AUDIO_X1, and AUDIO_CLK Clock Input Timing

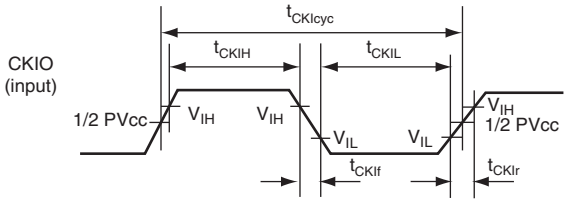


Figure 29.2 CKIO Clock Input Timing

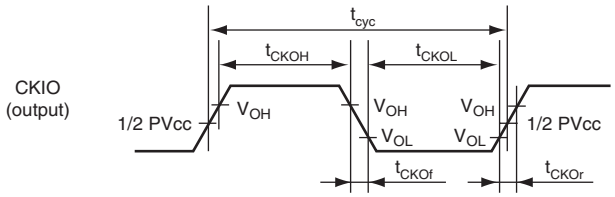
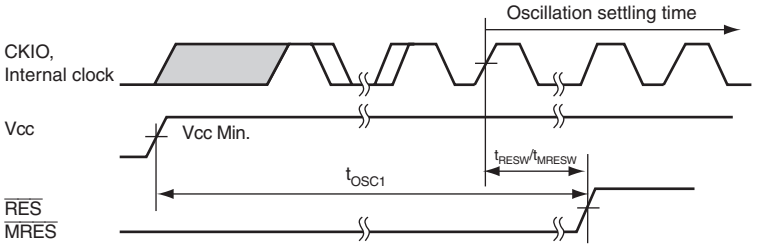
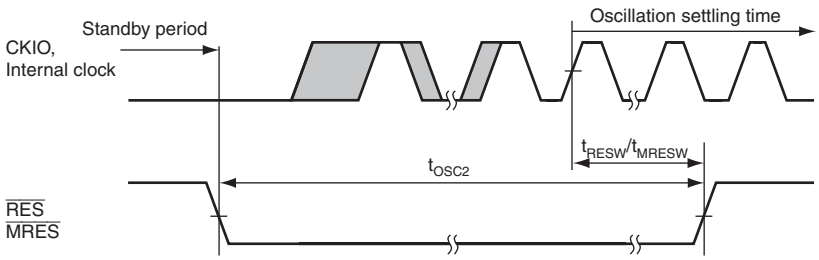


Figure 29.3 CKIO Clock Output Timing



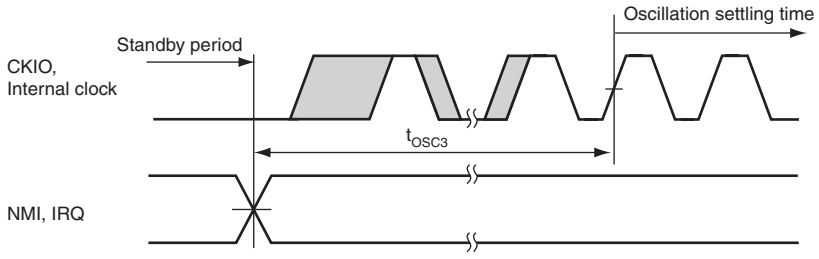
Note: Oscillation settling time when the internal oscillator is used.

Figure 29.4 Power-On Oscillation Settling Time



Note: Oscillation settling time when the internal oscillator is used.

Figure 29.5 Oscillation Settling Time on Return from Standby (Return by Reset)



Note: Oscillation settling time when the internal oscillator is used.

Figure 29.6 Oscillation Settling Time on Return from Standby (Return by NMI or IRQ)

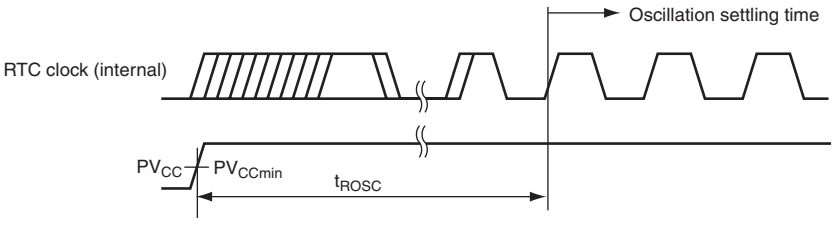


Figure 29.7 RTC Clock Oscillation Settling Time

29.3.2 Control Signal Timing

Table 29.6 Control Signal Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

		Bϕ = 60 MHz			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
$\overline{\text{RES}}$ pulse width	t_{RESW}	20*[2]	—	t_{cyc}	Figures 29.4, 29.5, and 29.8
$\overline{\text{RES}}$ setup time*[1]	t_{RESS}	200	—	ns	
$\overline{\text{MRES}}$ pulse width	t_{MRESW}	20*[3]	—	t_{cyc}	
$\overline{\text{MRES}}$ setup time*[1]	t_{MRESS}	200	—	ns	
NMI pulse width	t_{NMIW}	20*[4]	—	t_{cyc}	Figures 29.6, 29.9
NMI setup time*[1]	t_{NMIS}	150	—	ns	
NMI hold time	t_{NMIH}	10	—	ns	
IRQ7 to IRQ0 pulse width	t_{IRQW}	20*[4]	—	t_{cyc}	
IRQ7 to IRQ0 setup time*[1]	t_{IRQS}	150	—	ns	
IRQ7 to IRQ0 hold time	t_{IRQH}	10	—	ns	
PINT7 to PINT0 setup time*[1]	t_{PINTS}	150	—	ns	

Notes: 1. The $\overline{\text{RES}}$, $\overline{\text{MRES}}$, NMI, IRQ7 to IRQ0 and PINT7 to PINT0 signals are asynchronous signals. When the setup time is satisfied, change of signal level is detected at the rising edge of the clock. If not, the detection can be delayed until the rising edge of the next clock.
2. In software standby mode, deep standby mode or when the clock multiplication ratio is changed, $t_{RESW} = t_{OSC2}$ (min).
3. In software standby mode or deep standby mode, $t_{MRESW} = t_{OSC2}$ (min).
4. In software standby mode or deep standby mode, $t_{NMIW}/t_{IRQW} = t_{OSC3}$ (min).

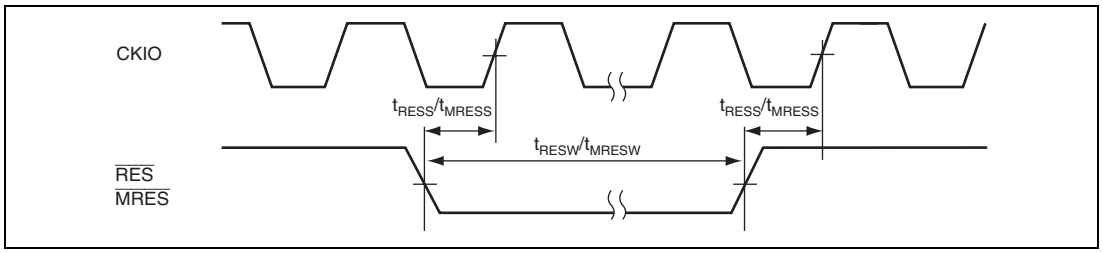


Figure 29.8 Reset Input Timing

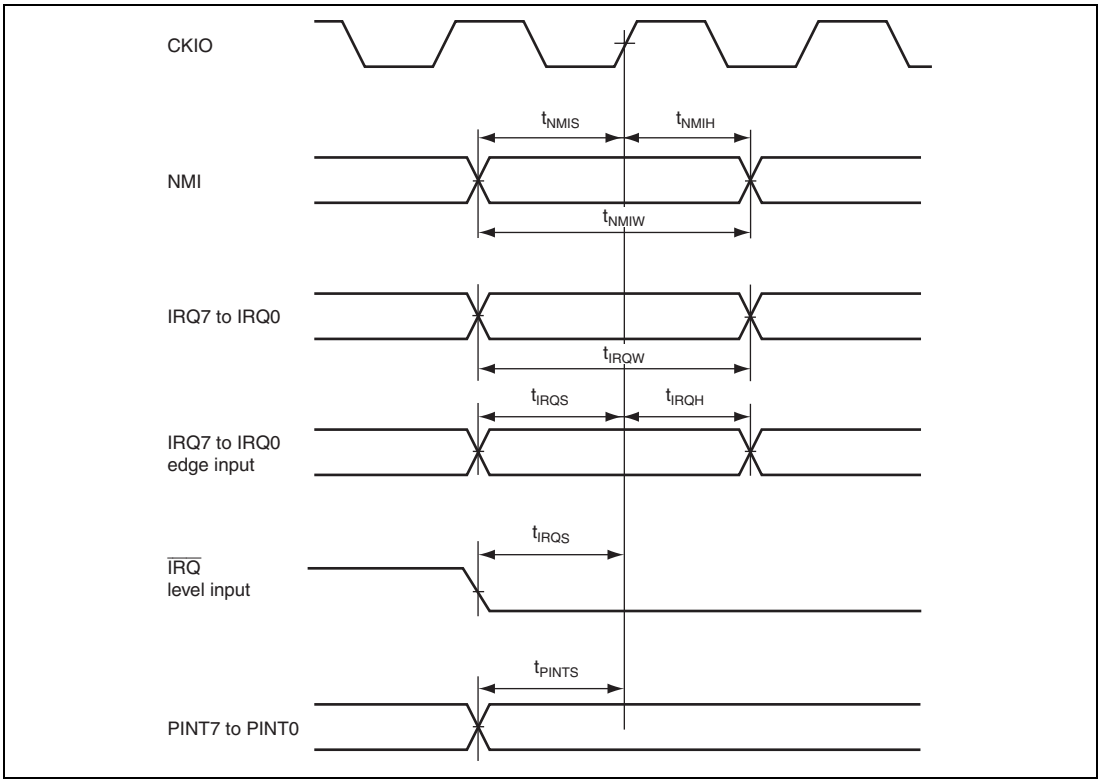


Figure 29.9 Interrupt Signal Input Timing

29.3.3 Bus Timing

Table 29.7 Bus Timing[1]

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

		Bϕ = 60 MHz[2]			
Item	**Symbol**	**Min.**	**Max.**	**Unit**	**Figure**
Address delay time 1 (external space)	t_{AD1}	—	13	ns	Figures 29.10 to 29.14
Address delay time 2 (SDRAM space)	t_{AD2}	1	13	ns	Figures 29.15 to 29.21
Byte control delay time	t_{BCD}	—	13	ns	Figures 29.10 to 29.14
Chip select delay time 1 (external space)	t_{CSD1}	—	13	ns	Figures 29.10 to 29.14
Chip select delay time 2 (SDRAM space)	t_{CSD2}	1	13	ns	Figures 29.15 to 29.21
Read strobe delay time	t_{RSD}	—	13	ns	Figures 29.10 to 29.14
Read data setup time 1 (external space)	t_{RDS1}	13	—	ns	Figures 29.10 to 29.14
Read data setup time 2 (SDRAM space)	t_{RDS2}	8	—	ns	Figures 29.15 to 29.21
Read data hold time 1 (external space)	t_{RDH1}	0	—	ns	Figures 29.10 to 29.14
Read data hold time 2 (SDRAM space)	t_{RDH2}	2	—	ns	Figures 29.15 to 29.21
Write enable delay time 1 (external space)	t_{WED1}	—	13	ns	Figures 29.10 to 29.14
Write enable delay time 2 (SDRAM space)	t_{WED2}	1	13	ns	Figures 29.15 to 29.21
Write data delay time 1 (external space)	t_{WDD1}	—	13	ns	Figures 29.10 to 29.14
Write data delay time 2 (SDRAM space)	t_{WDD2}	—	13	ns	Figures 29.15 to 29.21
Write data hold time 1 (external space)	t_{WDH1}	1	—	ns	Figures 29.10 to 29.14
Write data hold time 2 (SDRAM space)	t_{WDH2}	1	—	ns	Figures 29.15 to 29.21

Item	Symbol	Bϕ = 60 MHz*2 Min.	Max.	Unit	Figure
External wait setup time	t_{WTS}	8	—	ns	Figure 29.14
External wait hold time	t_{WTH}	5	—	ns	Figure 29.14
SDRAS delay time	t_{RASD}	1	13	ns	Figures 29.15 to 29.21
SDCAS delay time	t_{CASD}	1	13	ns	Figures 29.15 to 29.21
DQM delay time	t_{DQMD}	1	13	ns	Figures 29.15 to 29.21
CKE delay time	t_{CKED}	1	13	ns	Figure 29.21

Notes: 1. When writing to the external address space or making SDRAM settings in power-on reset exception handling or cancellation of deep standby mode, be sure to set bits ACOSW[3:0] in ACSWR to B'0011 beforehand.

2. The maximum value (f_{max}) of Bϕ (bus clock) depends on the number of wait cycles and the system configuration of your board.

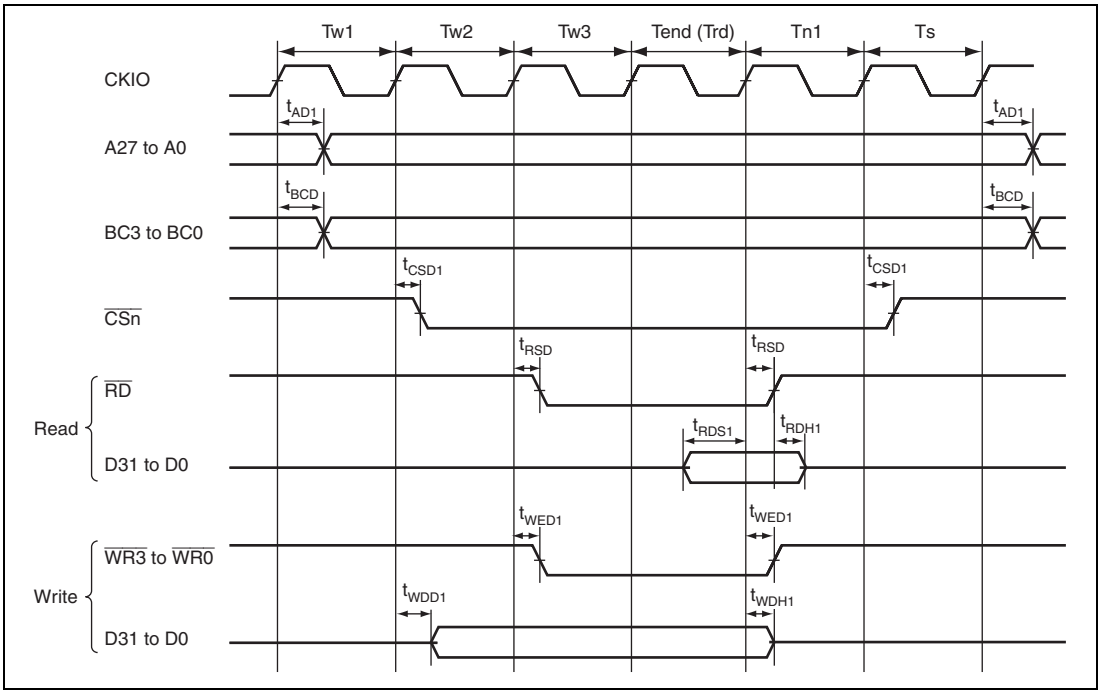


Figure 29.10 (1) External Address Space: Basic Bus Timing (Normal Access, Read/Write Cycle Wait = 3, CS Assert Wait = 1, Write Data Output Wait = 1, WR/RD Assert Wait = 2, Write Data Output Delay Cycles = 0, Read/Write CS Delay Cycles = 1)

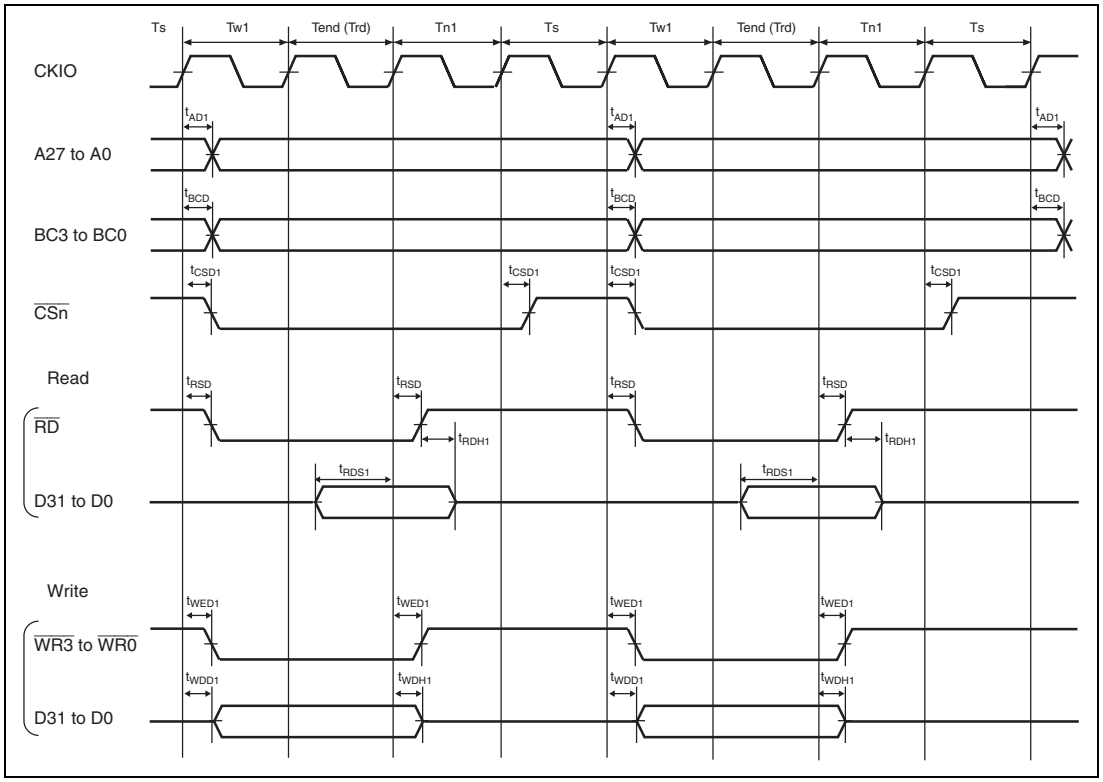


Figure 29.10 (2) External Address Space: Basic Bus Timing (Normal Access, Data Recovery Cycles = 0, Read/Write Cycle Wait = 1, Read/Write CS Delay Cycles = 1, Other Wait Settings = 0)

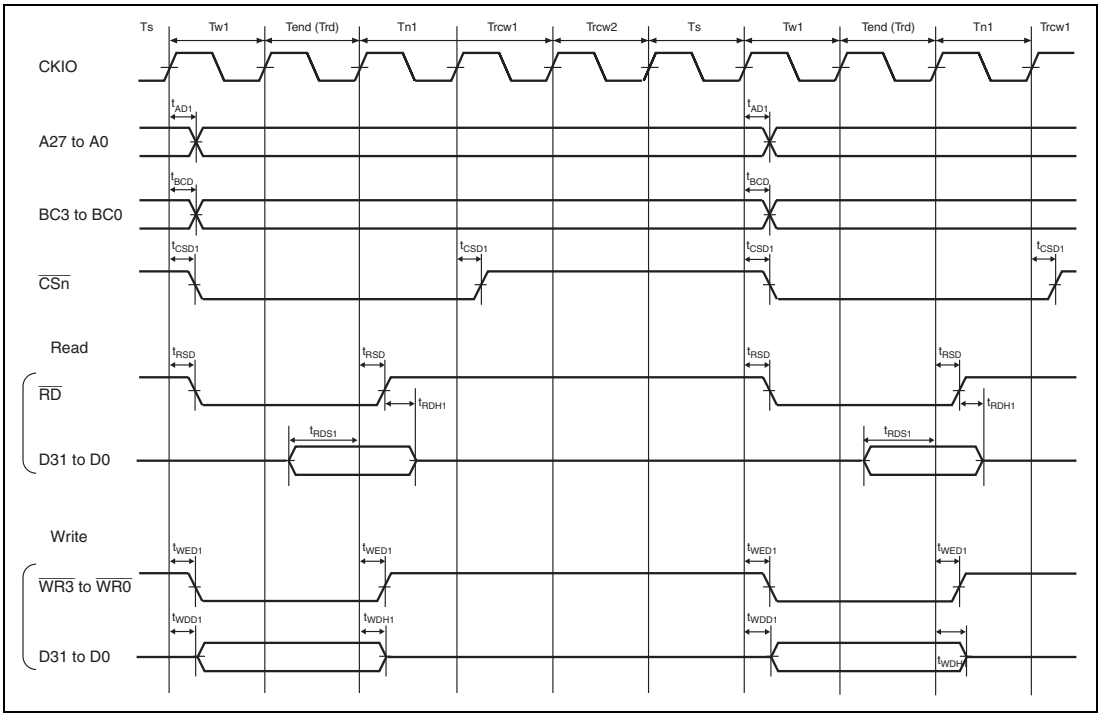


Figure 29.10 (3) External Address Space: Basic Bus Timing (Normal Access, Data Recovery Cycles = 2, Read/Write Cycle Wait = 1, Read/Write CS Delay Cycles = 1, Other Wait Settings = 0)

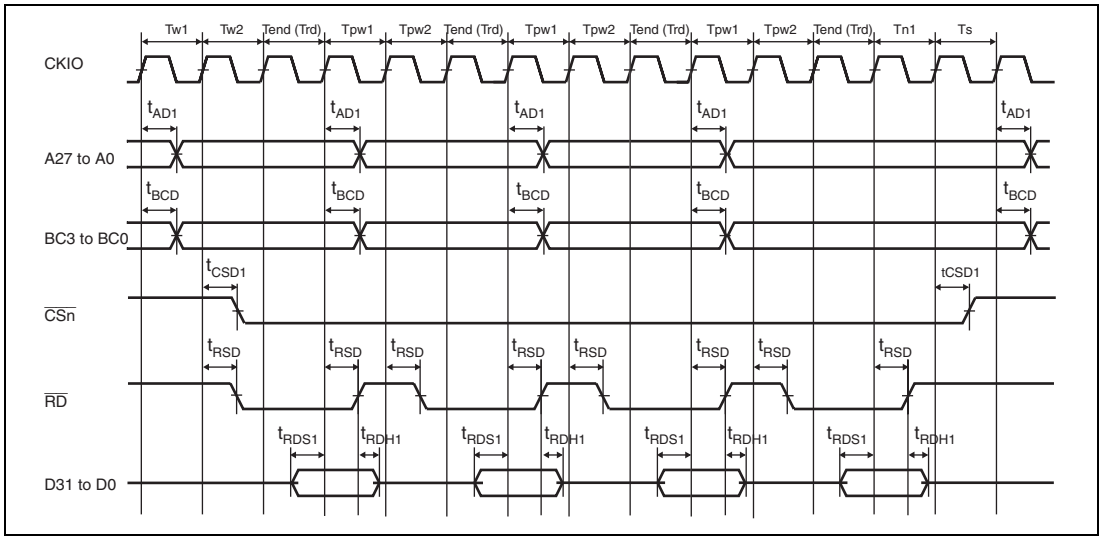


Figure 29.11 External Address Space: Basic Bus Timing (Page Read Access, Normal Access Compatible Mode , Read Cycle Wait = 2, Page Read Cycle Wait = 2, CS Assert Wait = 1, RD Assert Wait = 1, Read CS Delay Cycles = 1)

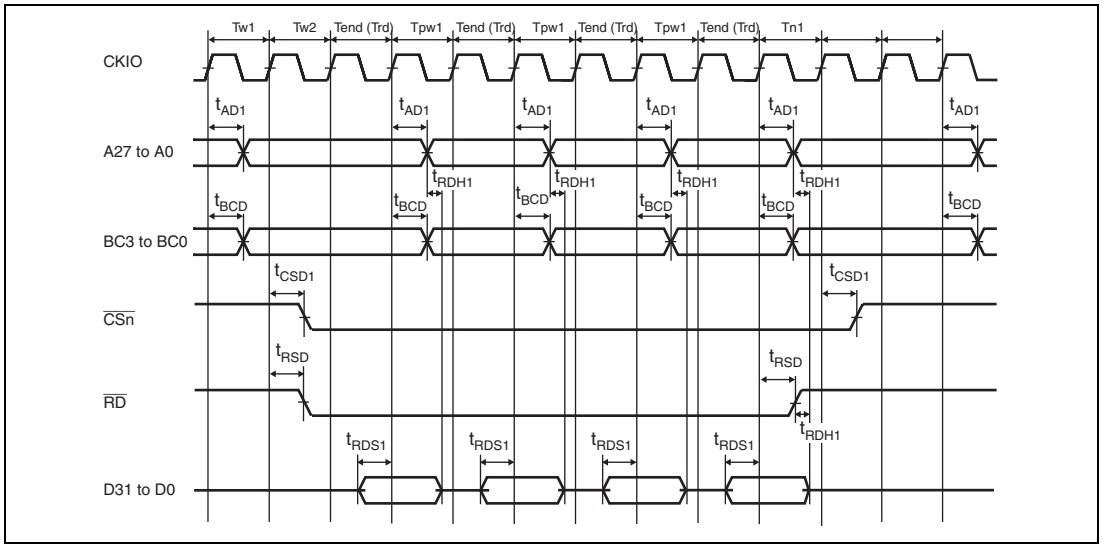


Figure 29.12 External Address Space: Basic Bus Timing (Page Read Access, External Read Data Continuous Assert Mode, Read Cycle Wait = 2, Page Read Cycle Wait = 1, CS Assert Wait = 1, RD Assert Wait = 1, Read CS Delay Cycles = 1)

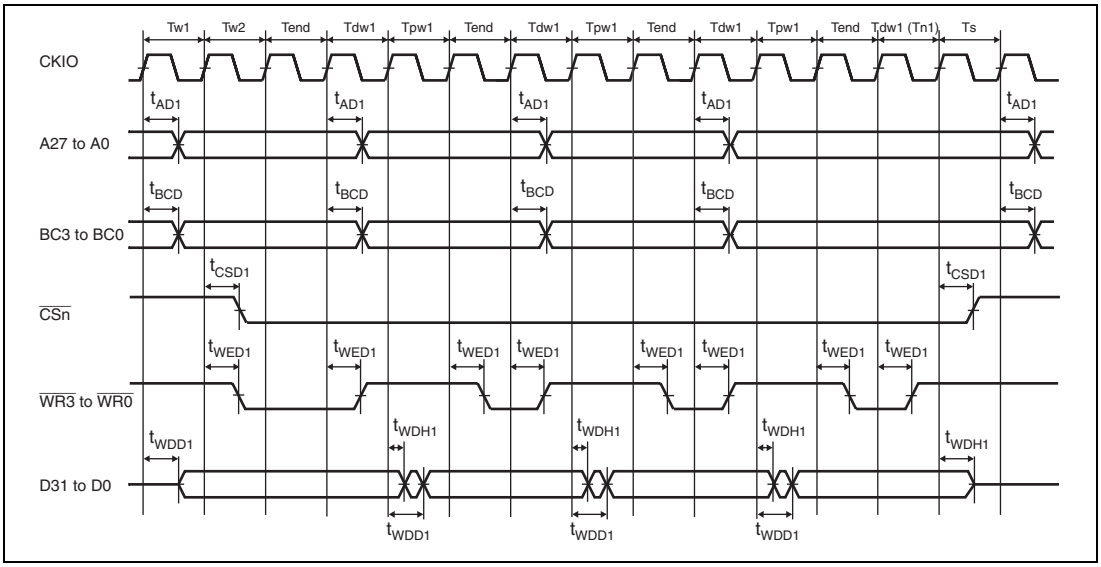


Figure 29.13 External Address Space: Basic Bus Timing (Page Write Access, Write Cycle Wait = 2, CS Assert Wait = 1, WR Assert Wait = 1, Write Data Output Delay Cycles = 1, Other Wait Settings = 0)

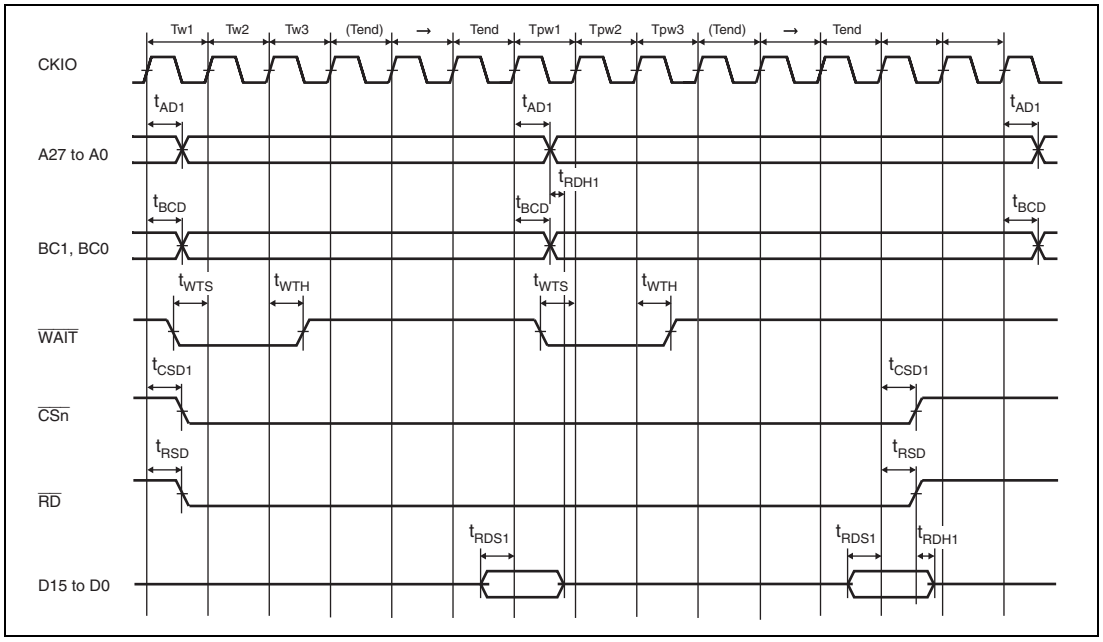
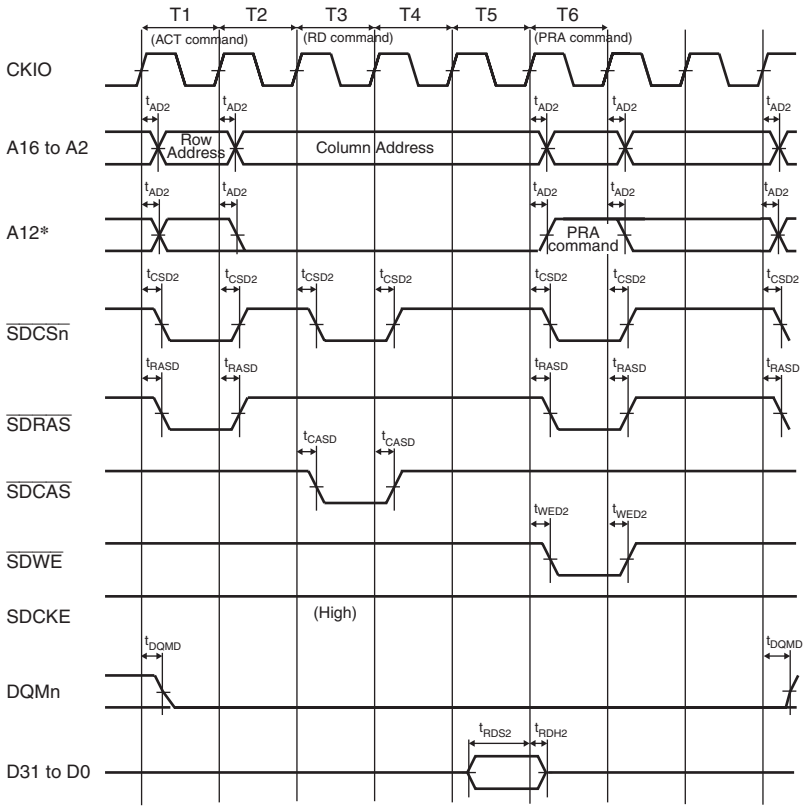


Figure 29.14 External Address Space: Timing with External Wait (Page Read Access to 16-Bit Width Channel, External Read Data Continuous Assert Mode, Read Cycle Wait = 3, Page Read Cycle Wait = 3, Other Wait Settings = 0, External Wait Cycles = 2)



Note: * Address pin connected to A10 in SDRAM.

Figure 29.15 Single Read Bus Timing for SDRAM Space (DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))

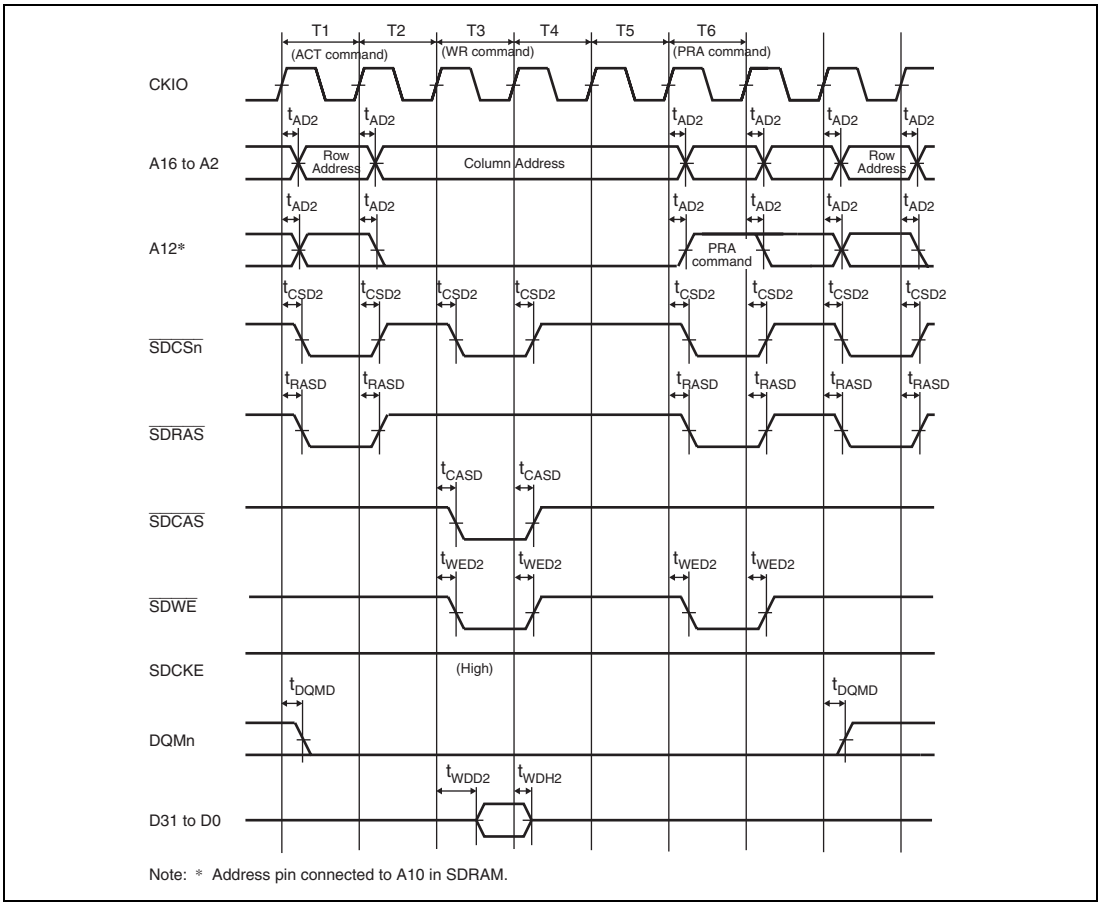


Figure 29.16 Single Write Bus Timing for SDRAM Space (DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))

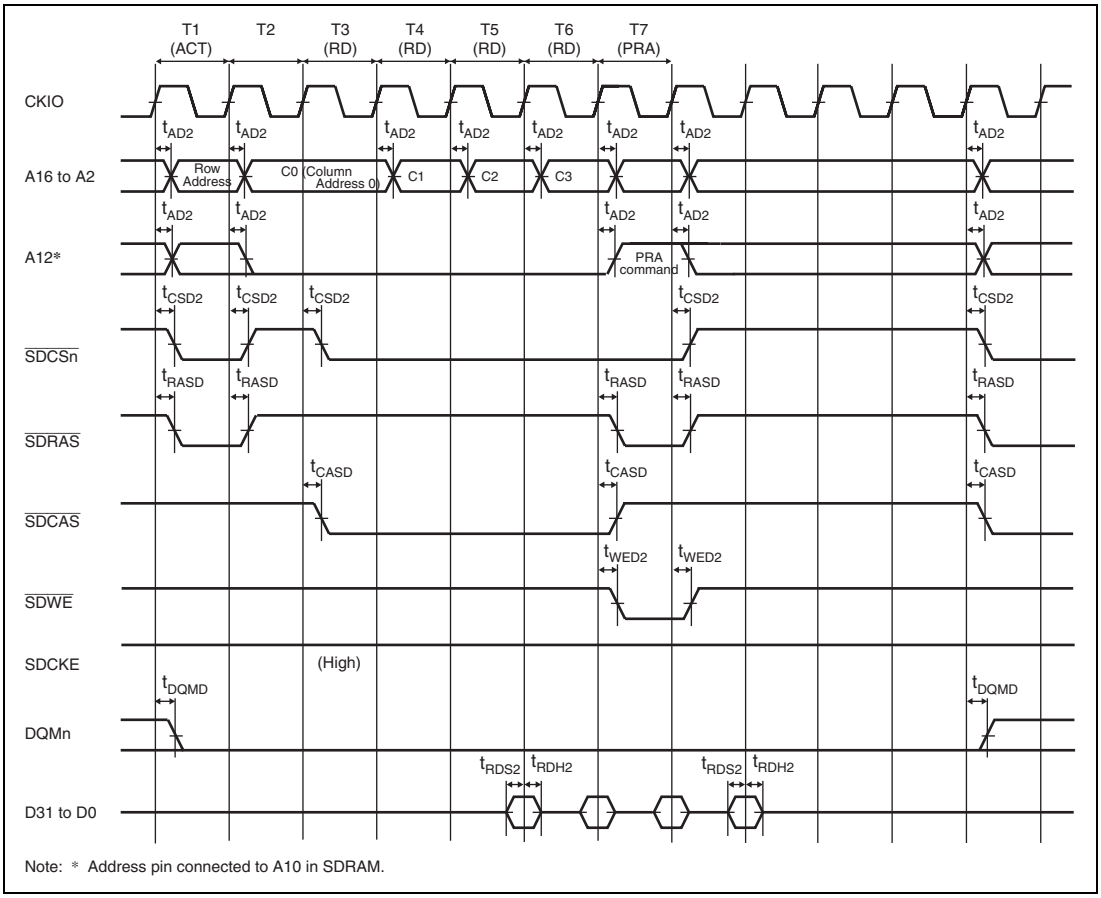
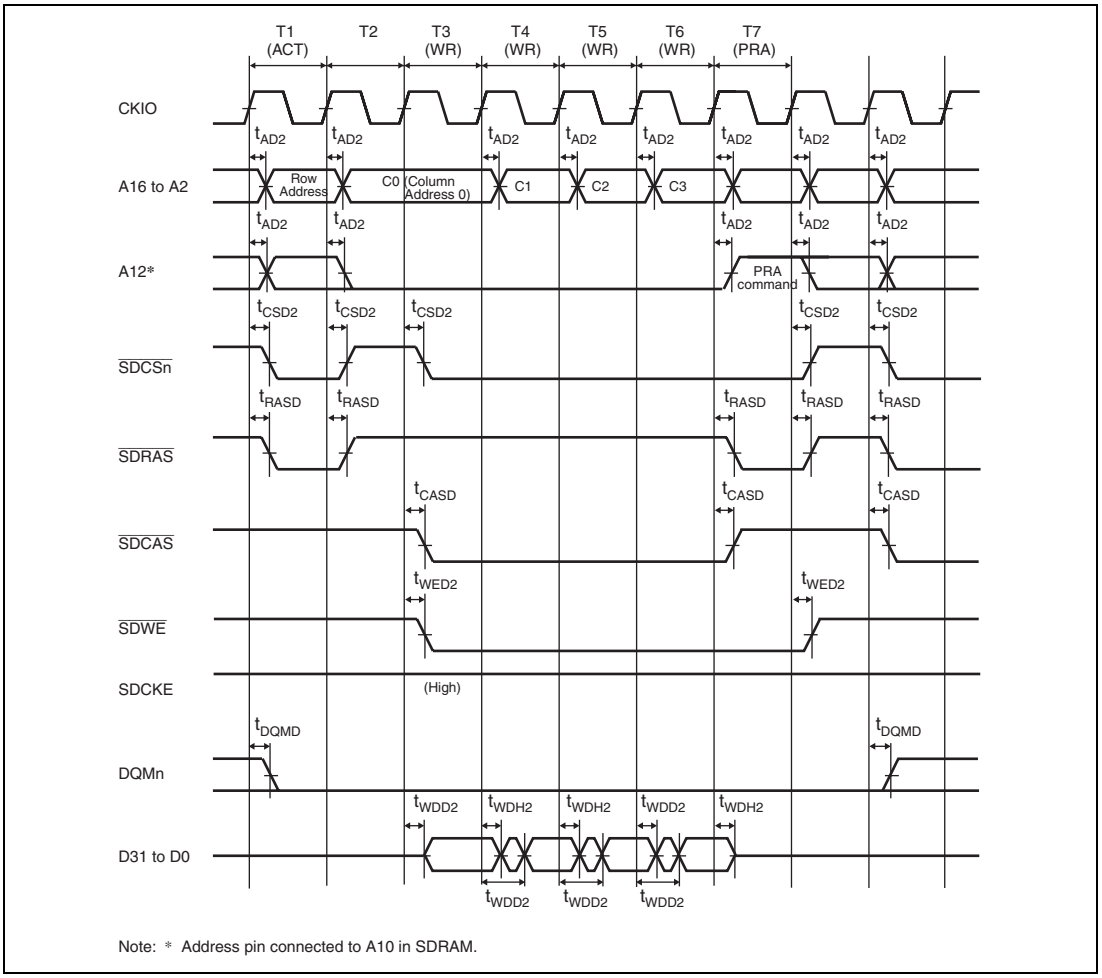
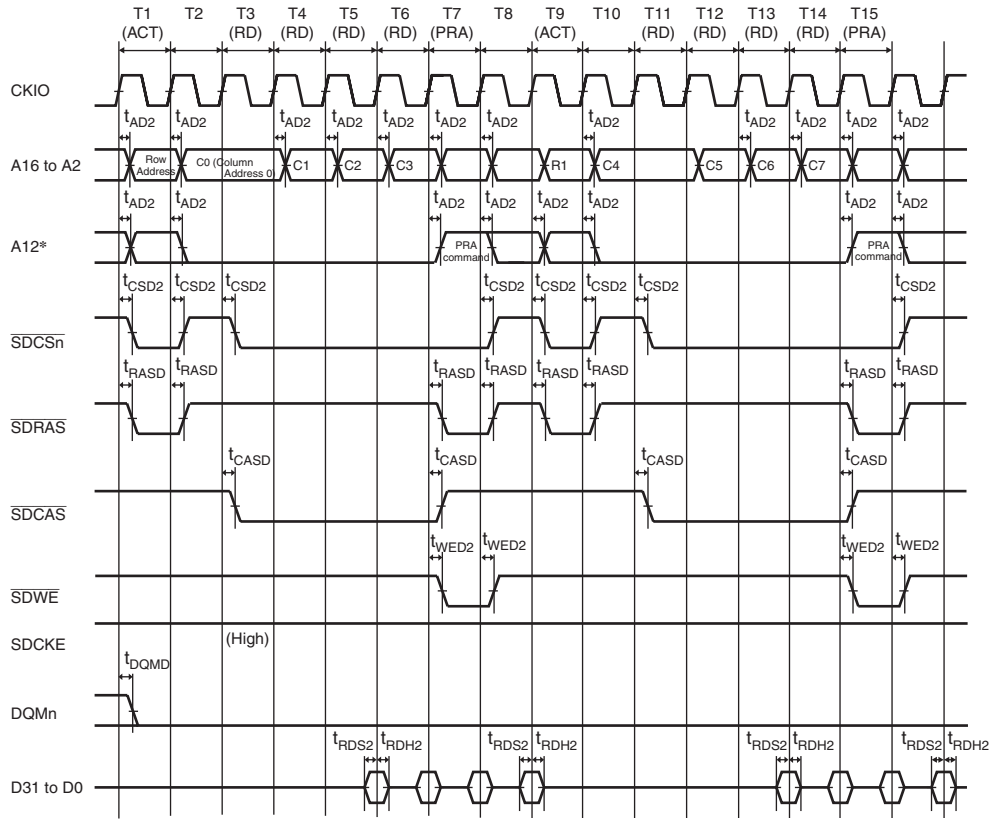


Figure 29.17 Multiple Read Bus Timing for SDRAM Space (Four Data Access, DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))



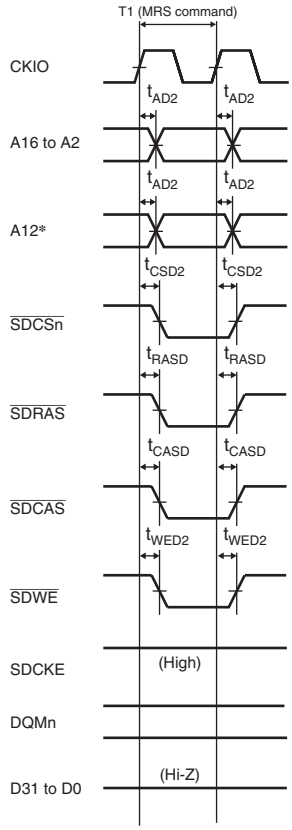
Note: * Address pin connected to A10 in SDRAM.

Figure 29.18 Multiple Write Bus Timing for SDRAM Space (Four Data Access, DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))



Note: * Address pin connected to A10 in SDRAM.

Figure 29.19 Multiple Read Row Span Bus Timing for SDRAM Space (Eight Data Access, DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))



Note: * Address pin connected to A10 in SDRAM.

Figure 29.20 Bus Timing for SDRAM Space Mode Register Setting

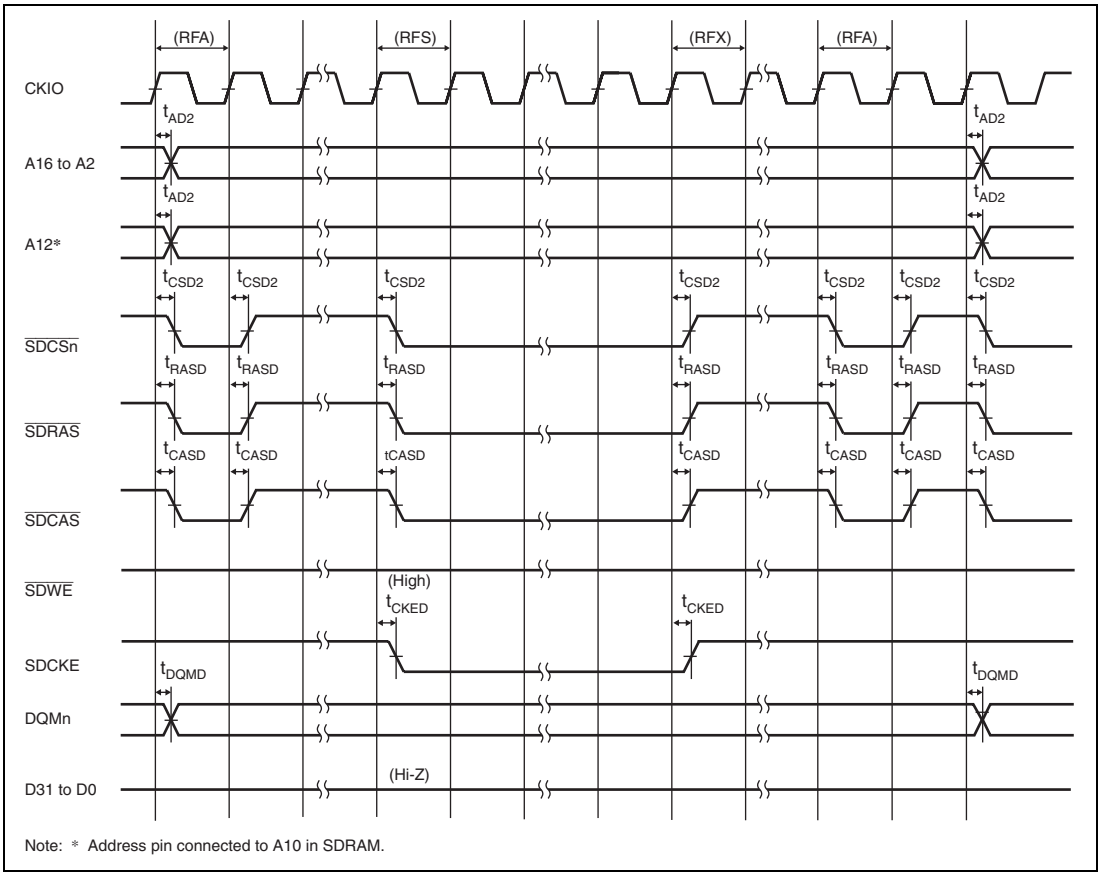


Figure 29.21 Bus Timing for SDRAM Space Self Refresh

29.3.4 DMAC Module Timing

Table 29.8 DMAC Module Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
DREQ setup time	t_{DRQS}	15	—	ns	Figure 29.22
DREQ hold time	t_{DRQH}	15	—		
DACK, DACT, DTEND delay time	t_{DACD}	—	15		Figure 29.23

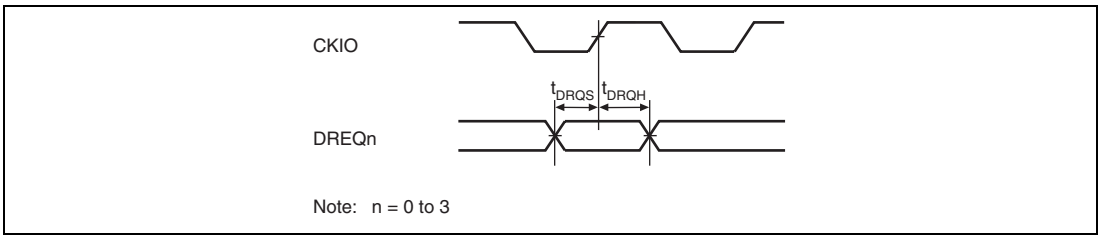


Figure 29.22 DREQ Input Timing

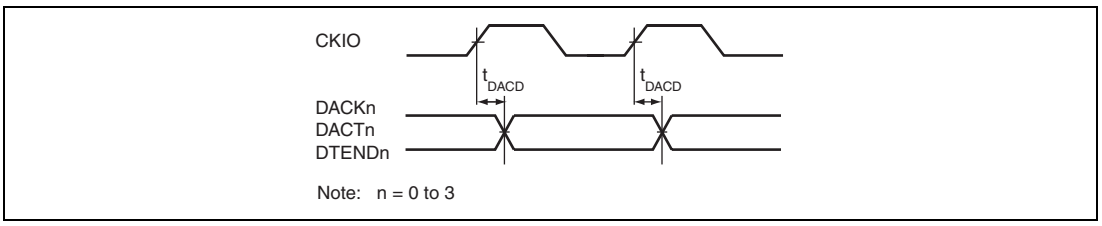


Figure 29.23 DACK, DACT, DTEND Output Timing

29.3.5 UBC Trigger Timing

Table 29.9 UBC Trigger Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{UBCTRG}}$ delay time	t_{UBCTGD}	—	14	ns	Figure 29.24

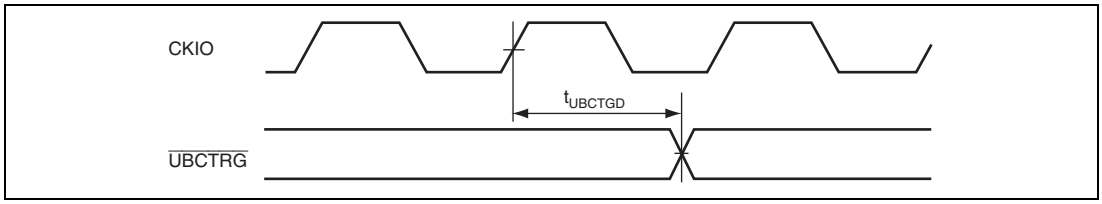


Figure 29.24 UBC Trigger Timing

29.3.6 MTU2 Module Timing

Table 29.10 MTU2 Module Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 29.25
Input capture input setup time	t_{TICS}	20	—	ns	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 29.26
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.

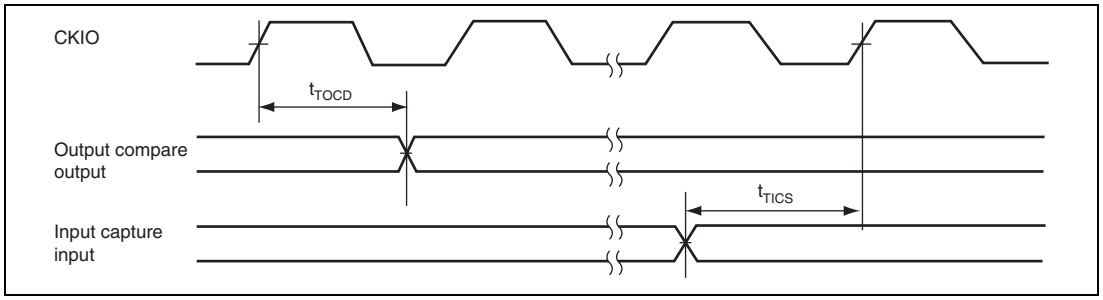


Figure 29.25 MTU2 Input/Output Timing

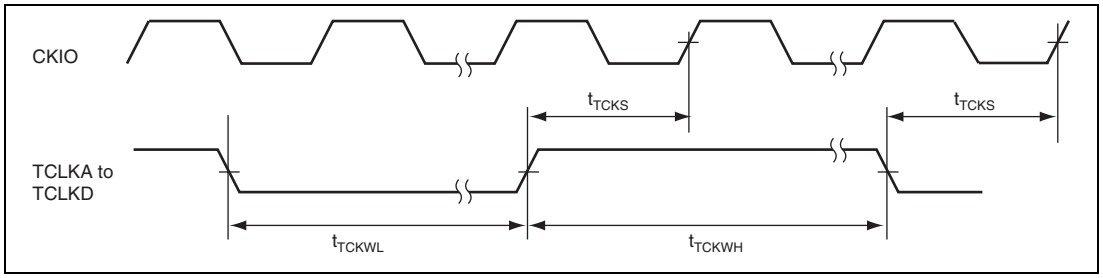


Figure 29.26 MTU2 Clock Input Timing

29.3.7 8-Bit Timer Timing

Table 29.11 8-Bit Timer Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Max.	Unit	Figure
Timer output delay time		t_{TMOD}	—	40	ns	Figure 29.27
Timer reset input setup time		t_{TMRS}	$(n - 1) \times t_{cyc} + 25$	—	ns	Figure 29.28
Timer clock input setup time		t_{TMCS}	$(n - 1) \times t_{cyc} + 25$	—	ns	Figure 29.29
Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{pcyc}	
	Both edges	t_{TMCWL}	2.5	—	t_{pcyc}	

Note: Above is the case in which the clock ratio B:P = n:1 (n = 1, 2, 3, 4, 6, 8, or 12)
t_{pcyc} indicates peripheral clock (Pϕ) cycle.

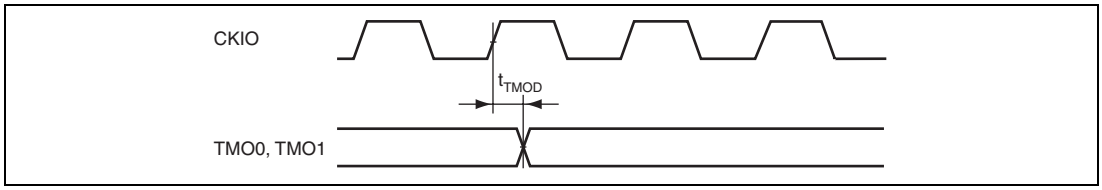


Figure 29.27 8-Bit Timer Output Timing

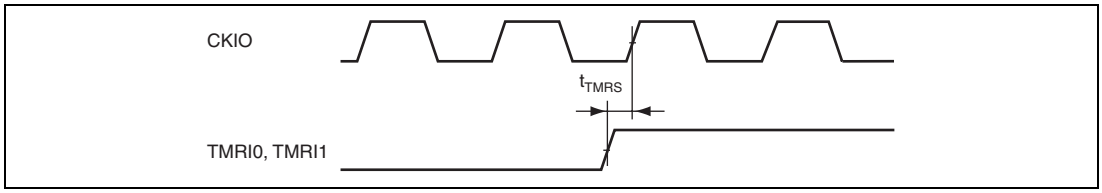


Figure 29.28 8-Bit Timer Reset Input Timing

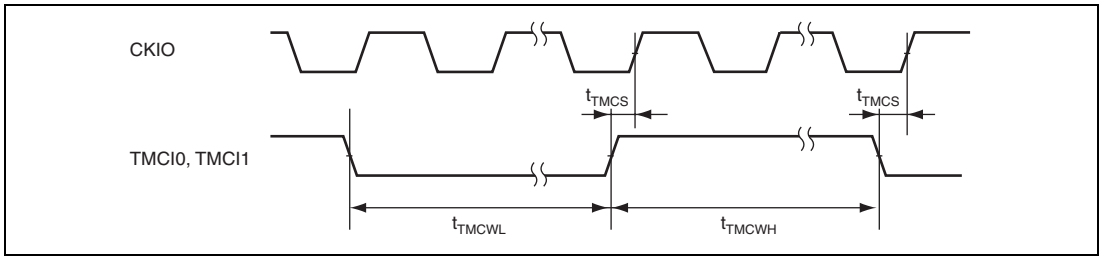


Figure 29.29 8-Bit Timer Clock Input Timing

29.3.8 Watchdog Timer Timing

Table 29.12 shows the timing of the watchdog timer.

Table 29.12 Watchdog Timer Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{WDTOVF}}$ delay time	t_{WOVD}	—	100	ns	Figure 29.30

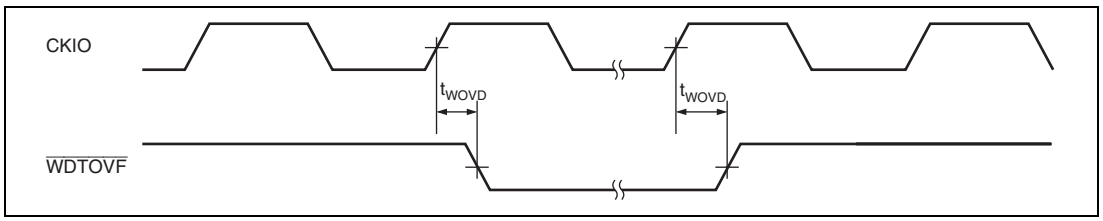


Figure 29.30 Watchdog Timer Timing

29.3.9 SCIF Module Timing

Table 29.13 SCIF Module Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item		Symbol	Min.	Max.	Unit	Figure
Input clock cycle	(Clocked synchronous)	t_{Scyc}	12	—	t_{pcyc}	Figure 29.31
	(Asynchronous)		4	—	t_{pcyc}	
Input clock rise time		t_{SCKr}	—	1.5	t_{pcyc}	
Input clock fall time		t_{SCKf}	—	1.5	t_{pcyc}	
Input clock width		t_{SCKW}	0.4	0.6	t_{Scyc}	
Transmit data delay time (Clocked synchronous)		t_{TXD}	—	$3\ t_{pcyc} + 15$	ns	Figure 29.32
Receive data setup time (Clocked synchronous)		t_{RXS}	$4\ t_{pcyc} + 15$	—	ns	
Receive data hold time (Clocked synchronous)		t_{RXH}	$1\ t_{pcyc} + 15$	—	ns	

Note: t_{pcyc} indicates a peripheral clock (Pϕ) cycle.

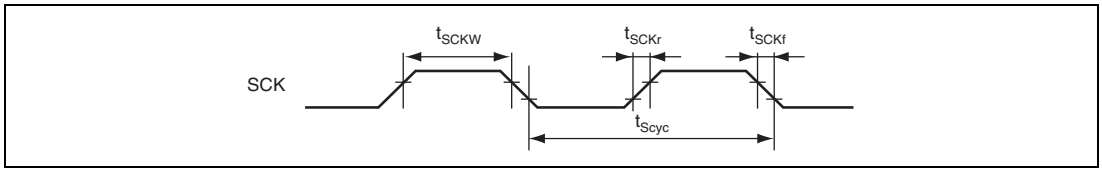


Figure 29.31 SCK Input Clock Timing

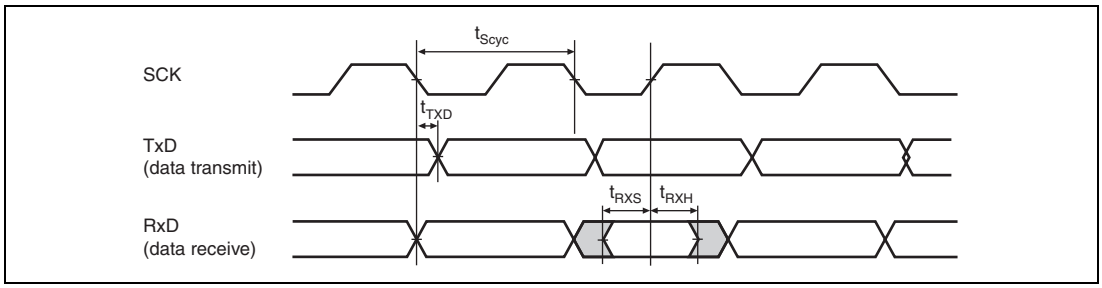


Figure 29.32 SCIF Input/Output Timing in Clocked Synchronous Mode

29.3.10 IIC3 Module Timing

Table 29.14 I²C Bus Interface 3 Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V ≤ $AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

			Specifications				
Item	**Symbol**	**Test Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**	**Figure**
SCL input cycle time	t_{SCL}		12 t_{pcyc} *1 + 600	—	—	ns	Figure 29.33
SCL input high pulse width	t_{SCLH}		3 t_{pcyc} *1 + 300	—	—	ns	
SCL input low pulse width	t_{SCLL}		5 t_{pcyc} *1 + 300	—	—	ns	
SCL, SDA input rise time	t_{Sr}		—	—	300	ns	
SCL, SDA input fall time	t_{Sf}		—	—	300	ns	
SCL, SDA input spike pulse removal time*2	t_{SP}		—	—	1.2	t_{pcyc} *1	
SDA input bus free time	t_{BUF}		5	—	—	t_{pcyc} *1	
Start condition input hold time	t_{STAH}		3	—	—	t_{pcyc} *1	
Retransmit start condition input setup time	t_{STAS}		3	—	—	t_{pcyc} *1	
Stop condition input setup time	t_{STOS}		3	—	—	t_{pcyc} *1	
Data input setup time	t_{SDAS}		1 t_{pcyc} *1 + 20	—	—	ns	
Data input hold time	t_{SDAH}		0	—	—	ns	
SCL, SDA capacitive load	Cb		0	—	400	pF	
SCL, SDA output fall time*3	t_{of}	PV_{CC} = 3.0 to 3.6 V	—	—	250	ns	

Notes: 1. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.
2. Depends on the value of NF2CYC.
3. Indicates the I/O buffer characteristics.

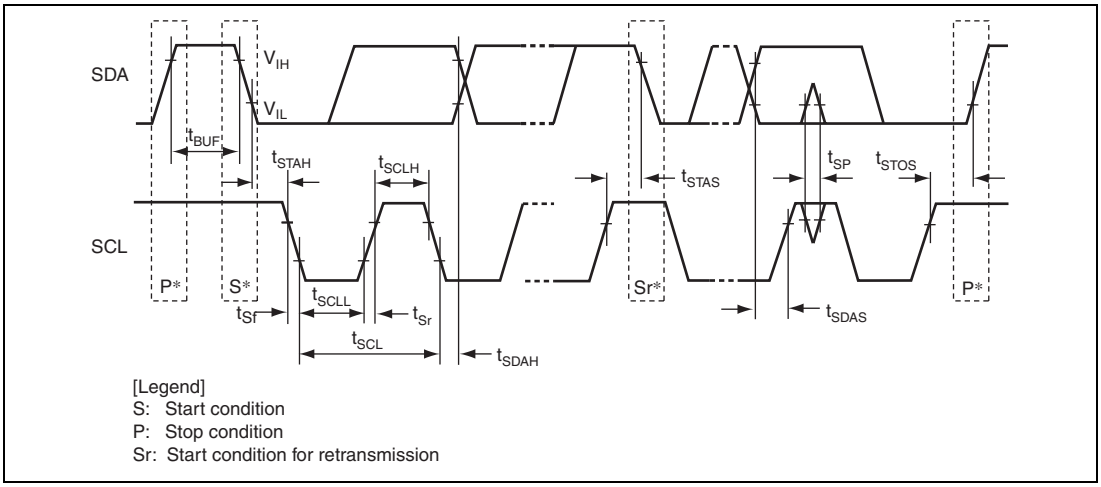


Figure 29.33 I²C Bus Interface 3 Input/Output Timing

29.3.11 SSI Module Timing

Table 29.15 SSI Module Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Typ.	Max.	Unit	Remarks	Figure
Output clock cycle	t_O	80	—	64000	ns	Output	Figure 29.34
Input clock cycle	t_I	80	—	64000	ns	Input	
Clock high	t_{HC}	32	—	—	ns	Bidirectional	
Clock low	t_{LC}	32	—	—	ns		
Clock rise time	t_{RC}	—	—	20	ns	Output (100 pF)	
Delay	t_{DTR}	—	—	50	ns	Transmit	Figures 29.35 and 29.36
Setup time	t_{SR}	15	—	—	ns	Receive	Figures 29.37 and 29.38
Hold time	t_{HTR}	5	—	—	ns	Receive	Figures 29.37 and 29.38
AUDIO_CLK input frequency	f_{AUDIO}	1	—	40	MHz		Figure 29.39

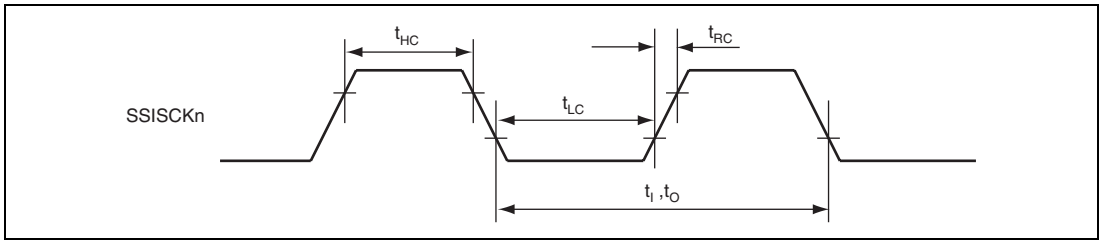


Figure 29.34 Clock Input/Output Timing

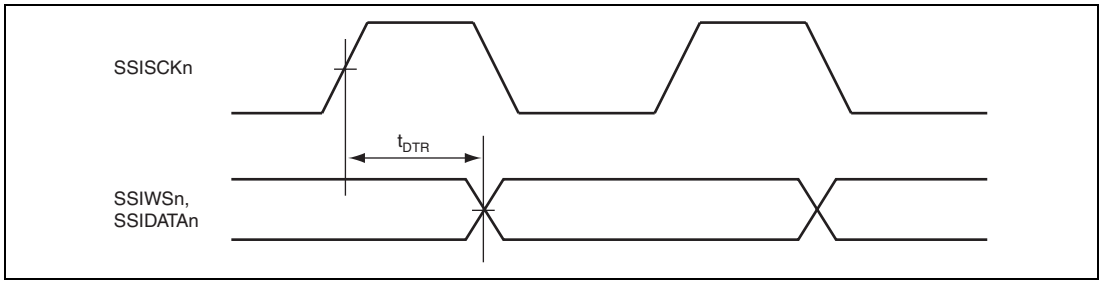


Figure 29.35 SSI Transmit Timing (1)

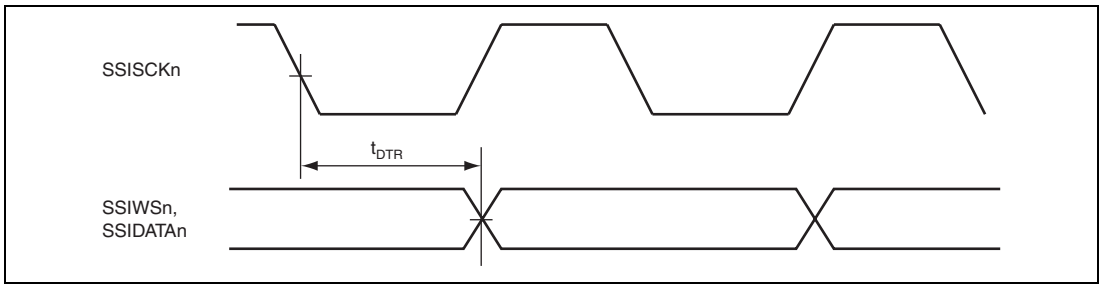


Figure 29.36 SSI Transmit Timing (2)

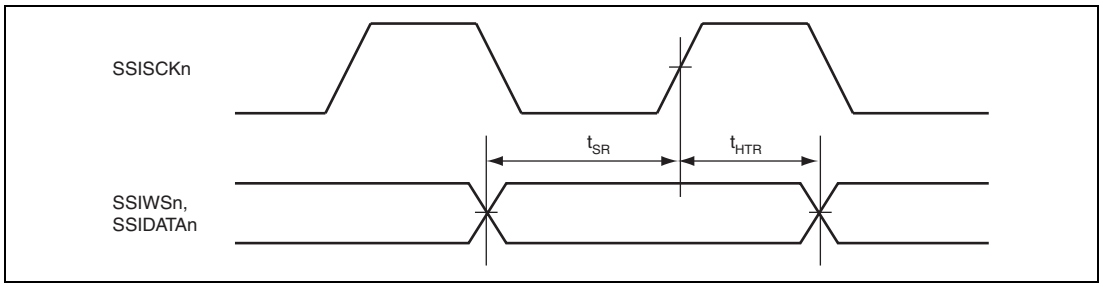


Figure 29.37 SSI Receive Timing (1)

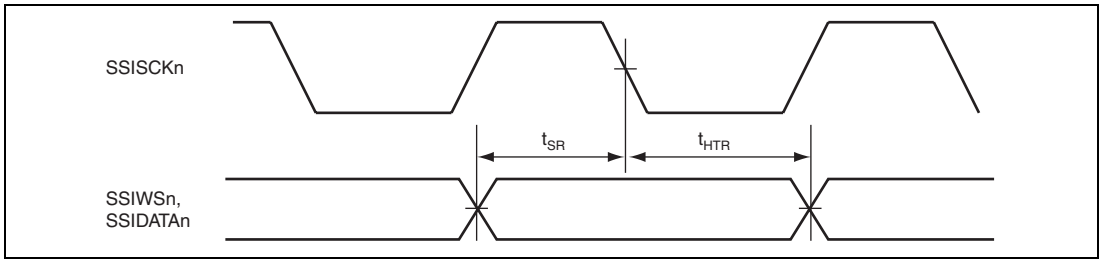


Figure 29.38 SSI Receive Timing (2)

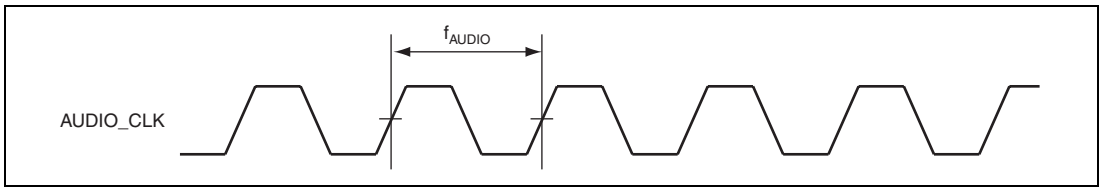


Figure 29.39 AUDIO_CLK Input Timing

29.3.12 RCAN-ET Module Timing

Table 29.16 RCAN-ET Module Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
Transmit data delay time	t_{CTXD}	—	100	ns	Figure 29.40
Receive data setup time	t_{CRXS}	100	—		
Receive data hold time	t_{CRXH}	100	—		

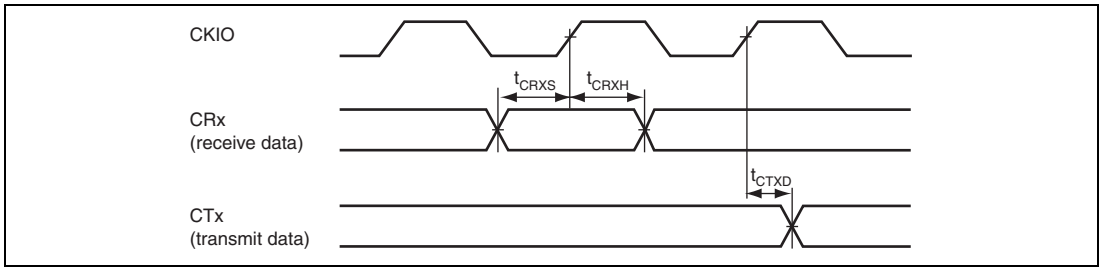


Figure 29.40 RCAN-ET Input/Output Timing

29.3.13 A/D Trigger Input Timing

Table 29.17 A/D Trigger Input Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Module	Item	Symbol	Min.	Max.	Unit	Figure
A/D converter	Trigger input setup time	t_{TRGS}	$(n - 1) \times t_{cyc} + 17$	—	ns	Figure 29.41

Note: Above is the case in which the clock ratio B:P = n:1 (n = 1, 2, 3, 4, 6, 8, or 12)

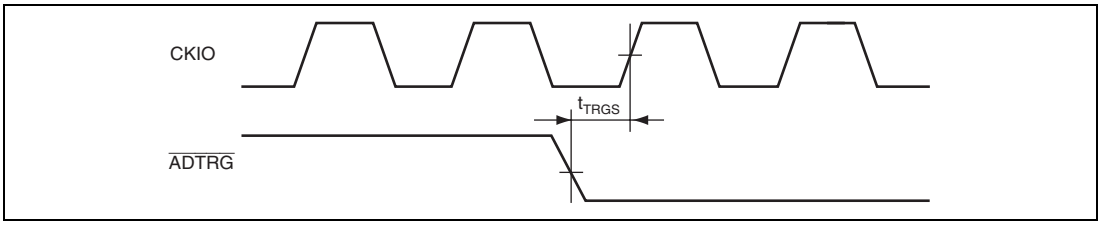


Figure 29.41 A/D Converter External Trigger Input Timing

29.3.14 I/O Port Timing

Table 29.18 I/O Port Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
Output data delay time	t_{PORTD}	—	100	ns	Figure 29.42
Input data setup time	t_{PORTS}	100	—		
Input data hold time	t_{PORTH}	100	—		

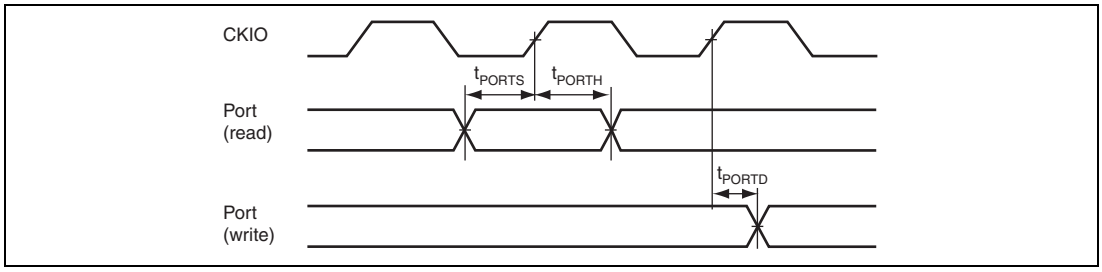


Figure 29.42 I/O Port Timing

29.3.15 H-UDI-Related Pin Timing

Table 29.19 H-UDI-Related Pin Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
UDTCK cycle time	t_{TCKcyc}	50*	—	ns	Figure 29.43
UDTCK high pulse width	t_{TCKH}	0.4	0.6	t_{TCKcyc}	
UDTCK low pulse width	t_{TCKL}	0.4	0.6	t_{TCKcyc}	
$\overline{\text{UDTRST}}$ pulse width	t_{TRSW}	20	—	t_{TCKcyc}	Figure 29.44
$\overline{\text{UDTRST}}$ setup time	t_{TRSS}	200	—	ns	
UDTDI setup time	t_{TDIS}	10	—	ns	Figure 29.45
UDTDI hold time	t_{TDIH}	10	—	ns	
UDTMS setup time	t_{TMSS}	10	—	ns	
UDTMS hold time	t_{TMSH}	10	—	ns	
UDTDO delay time	t_{TDOD}	—	16	ns	

Note: * Should be greater than the peripheral clock (Pϕ) cycle time.

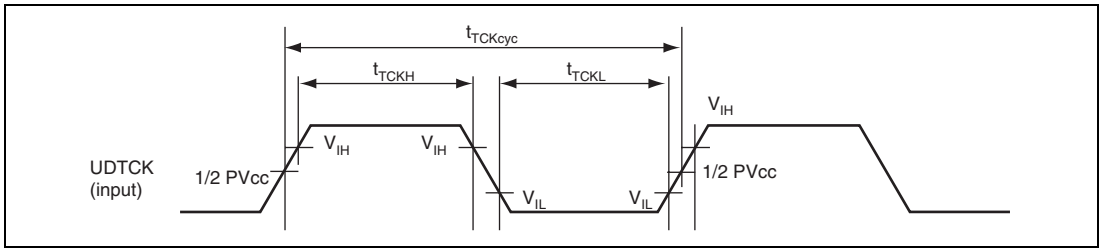


Figure 29.43 UDTCK Input Timing

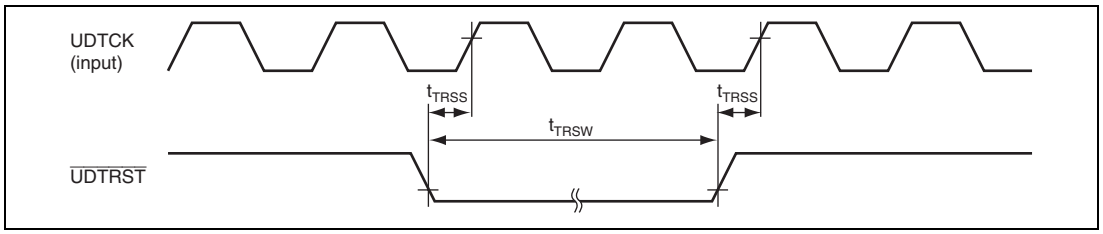


Figure 29.44 $\overline{\text{UDTRST}}$ Input Timing

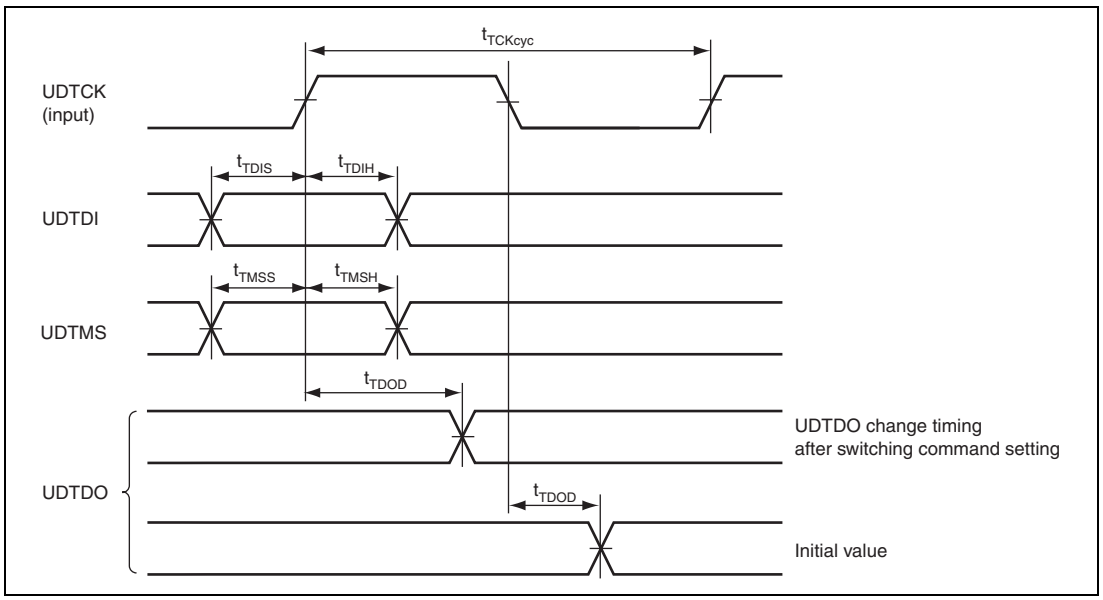


Figure 29.45 H-UDI Data Transfer Timing

29.3.16 AUD-II Timing

Table 29.20 AUD-II Timing

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Symbol	Min.	Max.	Unit	Figure
$\overline{\text{AUDRST}}$ pulse width	$t_{AUDRSTW}$	5	—	t_{RMCYC}	Figure 29.46
AUDMD setup time	t_{AUDMDS}	5	—	t_{RMCYC}	
RAM monitor clock cycle	t_{RMCYC}	33.33	—	ns	Figure 29.47
RAM monitor clock low pulse width	t_{RMCKWL}	0.4	0.6	t_{RMCYC}	
RAM monitor clock high pulse width	t_{RMCKWH}	0.4	0.6	t_{RMCYC}	
RAM monitor output data delay time	t_{RMDD}	2	14	ns	
RAM monitor input data setup time	t_{RMDS}	15	—	ns	
RAM monitor input data hold time	t_{RMDH}	5	—	ns	
RAM monitor SYNC setup time	t_{RMSS}	15	—	ns	
RAM monitor SYNC hold time	t_{RMSH}	5	—	ns	

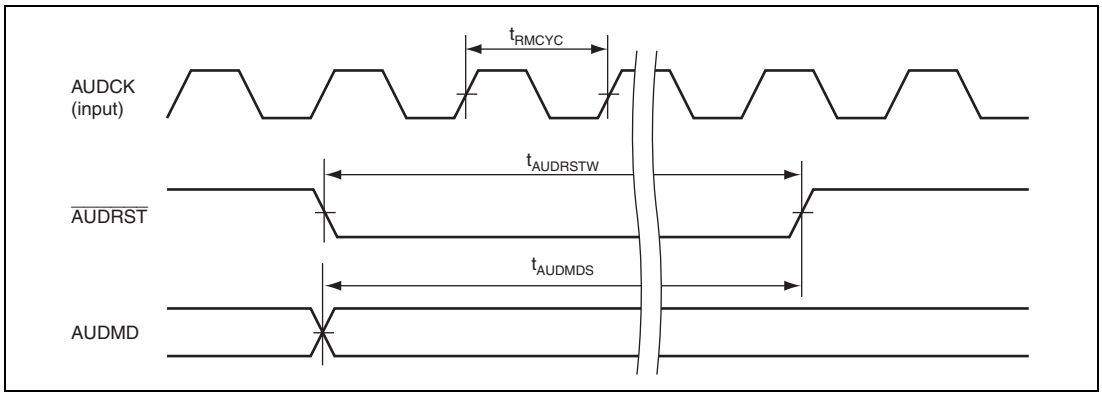


Figure 29.46 AUD Reset Timing

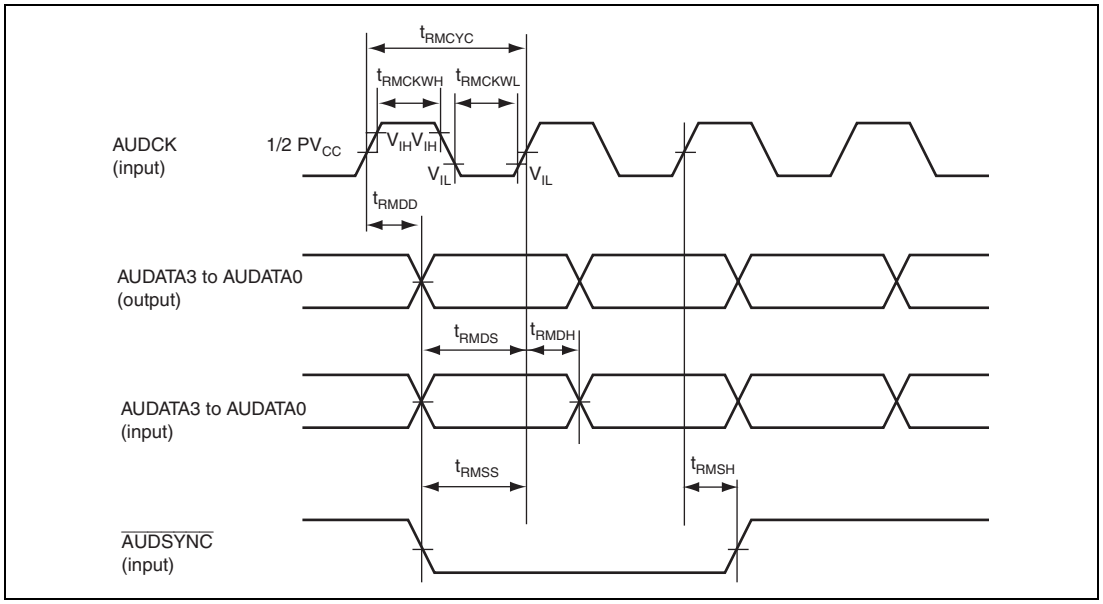
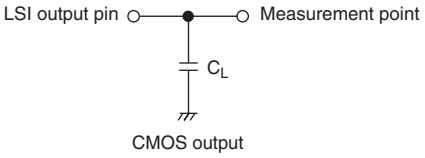


Figure 29.47 RAM Monitor Timing

29.3.17 AC Characteristics Measurement Conditions

- Input signal reference levels: high level = V_{IH} min, low level = V_{IL} max
- Output signal reference level: PVCC/2 (PVCC = 3.0 to 3.6 V)
- Input pulse level: PVSS to 3.0 V (where RES, MRES, NMI, MD1, MD0, MD_CLK1, MD_CLK0, ASEMD, UDTRST, and Schmitt trigger input pins are within PVSS to PVCC)
- Input rise and fall times: 1 ns



Note: C_L is the total value that includes the capacitance of measurement tools.
Each pin is set as follows:
30 pF: CKIO, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, $\overline{\text{CS0}}$ to $\overline{\text{CS6}}$, $\overline{\text{SDCS0}}$, $\overline{\text{SDCS1}}$, SDCKE, $\overline{\text{SDWE}}$, DQM0 to DQM3
50 pF: All other pins

Figure 29.48 Measurement Circuit

29.4 A/D Converter Characteristics

Table 29.21 lists the A/D converter characteristics.

Table 29.21 A/D Converter Characteristics

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	bits
Conversion time	3.9*2	—	—	μs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.0*1	LSB
Offset error	—	—	±2.0*1	LSB
Full-scale error	—	—	±2.0*1	LSB
Quantization error	—	—	±0.5*1	LSB
Absolute accuracy	—	—	±4.0	LSB

Notes: 1. Reference values
2. To satisfy the absolute accuracy, the conversion time should be 3.9 μs or longer.

29.5 D/A Converter Characteristics

Table 29.22 lists the D/A converter characteristics.

Table 29.22 D/A Converter Characteristics

Conditions: $PV_{CC} = V_{CC}R = PLLV_{CC} = 3.0$ V to 3.6 V, $AV_{CC} = 3.0$ V to 3.6 V,
$PV_{CC} - 0.3$ V $\leq AV_{CC} \leq PV_{CC}$, $AV_{ref} = 3.0$ V to AV_{CC},
$PV_{SS} = V_{SS}R = PLLV_{SS} = AV_{SS} = 0$ V

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	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

Note: * Reference values

29.6 Usage Note

Mount a multilayer ceramic capacitor between a pair of pins PVcc and PVss, VccR and VssR, or PLLVcc and PLLVss as a bypass capacitor. These capacitors must be placed as close as the power supply pins of the LSI. Also, a capacitor must be connected between the VCL and VSS pins to stabilize the power supply voltage that is internally lowered.

Figure 29.49 is an example of externally allocated capacitors.

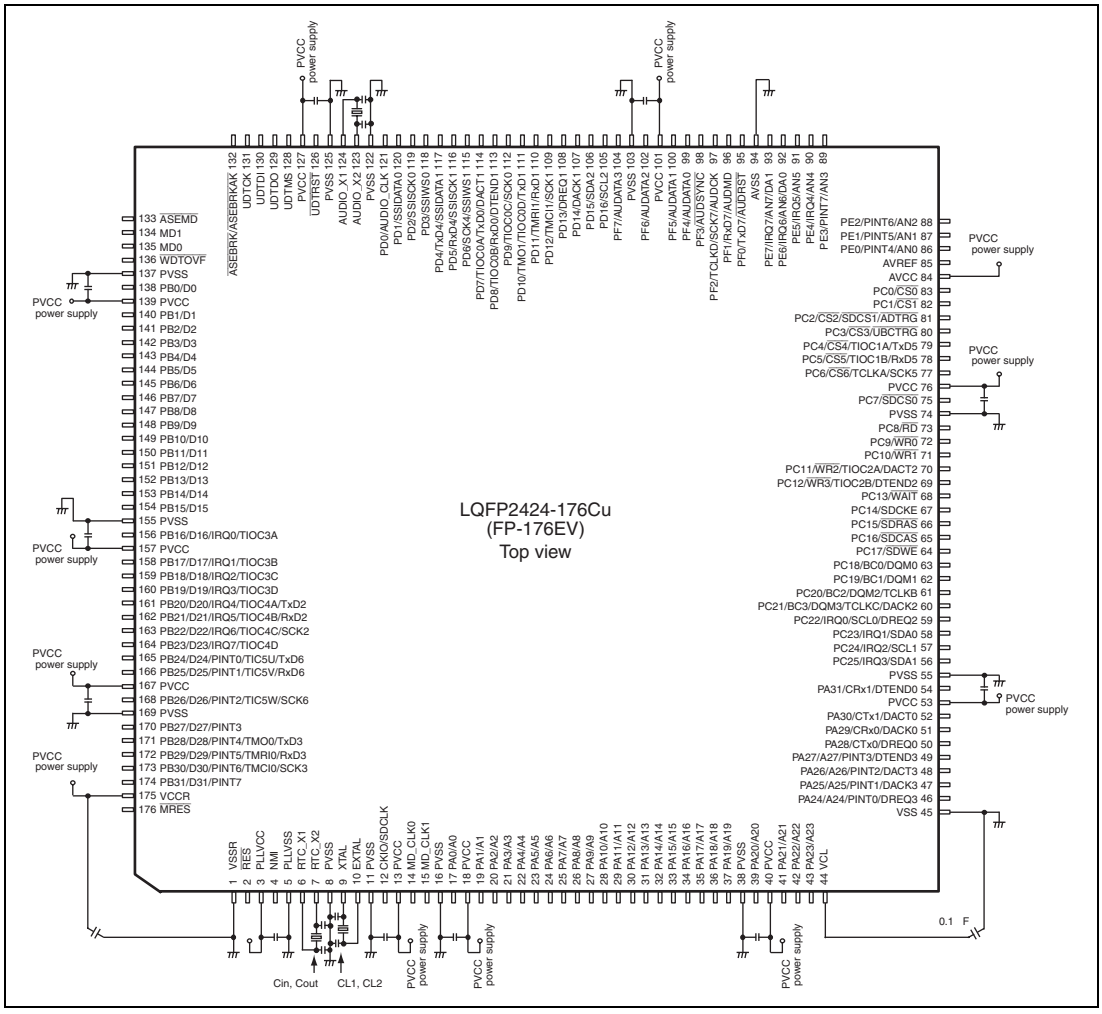


Figure 29.49 Example of Externally Allocated Capacitors

Appendix

A. Pin States

Table A.1 Pin States

Pin Function		Pin State						
		Reset State					Power-Down State	
		Power-On*2						
		Area 0 Data Bus Width						
Type	Pin Name	8 Bits	16 Bits	32 Bits	Manual	Sleep	Software Standby	Deep Standby
Clock	CKIO (clock modes 0 and 2)	O	O	O	O	O	L/Z*5	L/Z*5
	CKIO (clock mode 3)	I	I	I	I	I	I	I
	XTAL (clock modes 0 and 2)	O	O	O	O	O	L	L
	XTAL (clock mode 3)*1	O	O	O	O	O	L	L
	EXTAL (clock modes 0 and 2)	I	I	I	I	I	I	I
	EXTAL (clock mode 3)*1	Z	Z	Z	Z	Z	Z	Z
System control	$\overline{\text{RES}}$	I	I	I	I	I	I	I
	$\overline{\text{MRES}}$	—	—	—	I	I	I	I
	$\overline{\text{WDTOVF}}$	O	O	O	O	O	K	K
	$\overline{\text{ASEBRK}}$/$\overline{\text{ASEBRKAK}}$	H	H	H	O	O	I	K
Mode	MD1, MD0	I	I	I	I	I	I	I
	MD_CLK1, MD_CLK0	I	I	I	I	I	I	I
	$\overline{\text{ASEMD}}$	I	I	I	I	I	I	I
Interrupt	NMI	I	I	I	I	I	I	I
	IRQ7 to IRQ0	—	—	—	I	I	I	I*3
	PINT7 to PINT0	—	—	—	I	I	I	—

Pin Function		Pin State						
		Reset State					Power-Down State	
		Power-On*2						
		Area 0 Data Bus Width						
Type	Pin Name	8 Bits	16 Bits	32 Bits	Manual	Sleep	Software Standby	Deep Standby
Address data	A27 to A24	Z	Z	Z	O	O	K	K
	A23 to A0	L	L	L	O	O	K	K
	D31 to D16	—	—	Z	I/O	I/O	Z	K
	D15 to D8	—	Z	Z	I/O	I/O	Z	K
	D7 to D0	Z	Z	Z	I/O	I/O	Z	K
Bus control	$\overline{\text{WAIT}}$	—	—	—	I	I	Z	Z
	$\overline{\text{CS0}}$	H	H	H	O	O	K	K
	$\overline{\text{CS6}}$ to $\overline{\text{CS1}}$	—	—	—	O	O	K	K
	$\overline{\text{RD}}$	H	H	H	O	O	K	K
	$\overline{\text{WR3}}$	—	—	H	O	O	K	K
	$\overline{\text{WR2}}$	—	—	H	O	O	K	K
	$\overline{\text{WR1}}$	—	H	H	O	O	K	K
	$\overline{\text{WR0}}$	H	H	H	O	O	K	K
	BC3 to BC0	—	—	—	O	O	K	K
	$\overline{\text{SDCS1}}$, $\overline{\text{SDCS0}}$	—	—	—	O	O	K	K
	$\overline{\text{SDRAS}}$	—	—	—	O	O	K	K
	$\overline{\text{SDCAS}}$	—	—	—	O	O	K	K
	$\overline{\text{SDWE}}$	—	—	—	O	O	K	K
	DQM3 to DQM0	—	—	—	O	O	K	K
	SDCKE	—	—	—	O	O	K	K
DMAC	DREQ3 to DREQ0	—	—	—	I	I	Z	Z
	DACK3 to DACK0	—	—	—	O	O	K	K
	DACT3 to DACT0	—	—	—	O	O	K	K
	DTEND3 to DTEND0	—	—	—	O	O	K	K

Pin Function		Pin State						
		Reset State					Power-Down State	
		Power-On*2						
		Area 0 Data Bus Width						
Type	Pin Name	8 Bits	16 Bits	32 Bits	Manual	Sleep	Software Standby	Deep Standby
MTU2	TCLKA to TCLKD	—	—	—	I	I	Z	Z
	TIOC0A to TIOC0D	—	—	—	I/O	I/O	K	K
	TIOC1A, TIOC1B	—	—	—	I/O	I/O	K	K
	TIOC2A, TIOC2B	—	—	—	I/O	I/O	K	K
	TIOC3A to TIOC3D	—	—	—	I/O	I/O	K	K
	TIOC4A to TIOC4D	—	—	—	I/O	I/O	K	K
	TIC5U, TIC5V, TIC5W	—	—	—	I	I	Z	Z
TMR	TMO1, TMO0	—	—	—	O	O	K	K
	TMCI1, TMCI0	—	—	—	I	I	Z	Z
	TMRI1, TMRI0	—	—	—	I	I	Z	Z
SCIF	SCK7 to SCK0	—	—	—	I/O	I/O	K	K
	RxD7 to RxD0	—	—	—	I	I	Z	Z
	TxD7 to TxD0	—	—	—	O	O	K	K
IIC3	SCL2 to SCL0	—	—	—	I/O	I/O	Z	Z
	SDA2 to SDA0	—	—	—	I/O	I/O	Z	Z
SSI	SSIDATA1, SSIDATA0	—	—	—	I/O	I/O	K	K
	SSISCK1, SSISCK0	—	—	—	I/O	I/O	K	K
	SSIWS1, SSIWS0	—	—	—	I/O	I/O	K	K
	AUDIO_CLK	—	—	—	I	I	Z	Z
	AUDIO_X1*1	I	I	I	I	I	I	I
	AUDIO_X2*1	O	O	O	O	O	O	O
RCAN_ET	CRx1, CRx0	—	—	—	I	I	Z	Z
	CTx1, CTx0	—	—	—	O	O	K	K

Pin Function		Pin State						
		Reset State					Power-Down State	
		Power-On*2						
		Area 0 Data Bus Width						
Type	Pin Name	8 Bits	16 Bits	32 Bits	Manual	Sleep	Software Standby	Deep Standby
A/D converter	AN7 to AN0	—	—	—	I	I	Z	Z
	$\overline{\text{ADTRG}}$	—	—	—	I	I	Z	Z
D/A converter	DA1, DA0	—	—	—	O	O	O	Z
RTC	RTC_X1*1	I	I	I	I	I	I	I
	RTC_X2*1	O	O	O	O	O	O	O
AUD-II	$\overline{\text{AUDRST}}$	—	—	—	I	I	I	Z
	AUDMD	—	—	—	I	I	I	Z
	$\overline{\text{AUDSYNC}}$	—	—	—	I/O	I/O	I/O	K
	AUDCK	—	—	—	I/O	I/O	I/O	K
	AUDATA3 to AUDATA0	—	—	—	I/O	I/O	I/O	K
H-UDI	UDTCK	I	I	I	I	I	I	I
	UDTMS	I	I	I	I	I	I	I
	UDTDI	I	I	I	I	I	I	I
	UDTDO	O/Z*4	O/Z*4	O/Z*4	O/Z*4	O/Z*4	O/Z*4	K
	$\overline{\text{UDTRST}}$	I	I	I	I	I	I	I
UBC	$\overline{\text{UBCTRG}}$	—	—	—	O	O	O	K
I/O ports	PA31 to PA28	I	I	I	I/O	I/O	K	K
	PA27 to PA0	L	L	L	I/O	I/O	K	K
	PB31 to PB16	I	I	Z	I/O	I/O	K	K
	PB15 to PB8	I	Z	Z	I/O	I/O	K	K
	PB7 to PB0	Z	Z	Z	I/O	I/O	K	K

Pin Function		Pin State						
		Reset State					Power-Down State	
		Power-On*2						
		Area 0 Data Bus Width						
Type	Pin Name	8 Bits	16 Bits	32 Bits	Manual	Sleep	Software Standby	Deep Standby
I/O ports	PC25 to PC22	I	I	I	I	I	Z	Z
	PC21 to PC13	I	I	I	I/O	I/O	K	K
	PC12, PC11	I	I	H	I/O	I/O	K	K
	PC10	I	H	H	I/O	I/O	K	K
	PC9	H	H	H	I/O	I/O	K	K
	PC8	H	H	H	I/O	I/O	K	K
	PC7 to PC1	I	I	I	I/O	I/O	K	K
	PC0	H	H	H	I/O	I/O	K	K
	PD16, PD15	I	I	I	I	I	Z	Z
	PD14 to PD0	I	I	I	I/O	I/O	K	K
	PE7 to PE0	I	I	I	I	I	Z	Z
	PF7 to PF0	I	I	I	I/O	I/O	K	K

[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. When pins for the connection with a crystal resonator are not used, the EXTAL and AUDIO_X1 pins must be pulled up and the XTAL and AUDIO_X2 pins must be open. The RTC_X1 pin must be connected to GND and the RTC_X2 must be open.

2. 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 23, Pin Function Controller (PFC)).

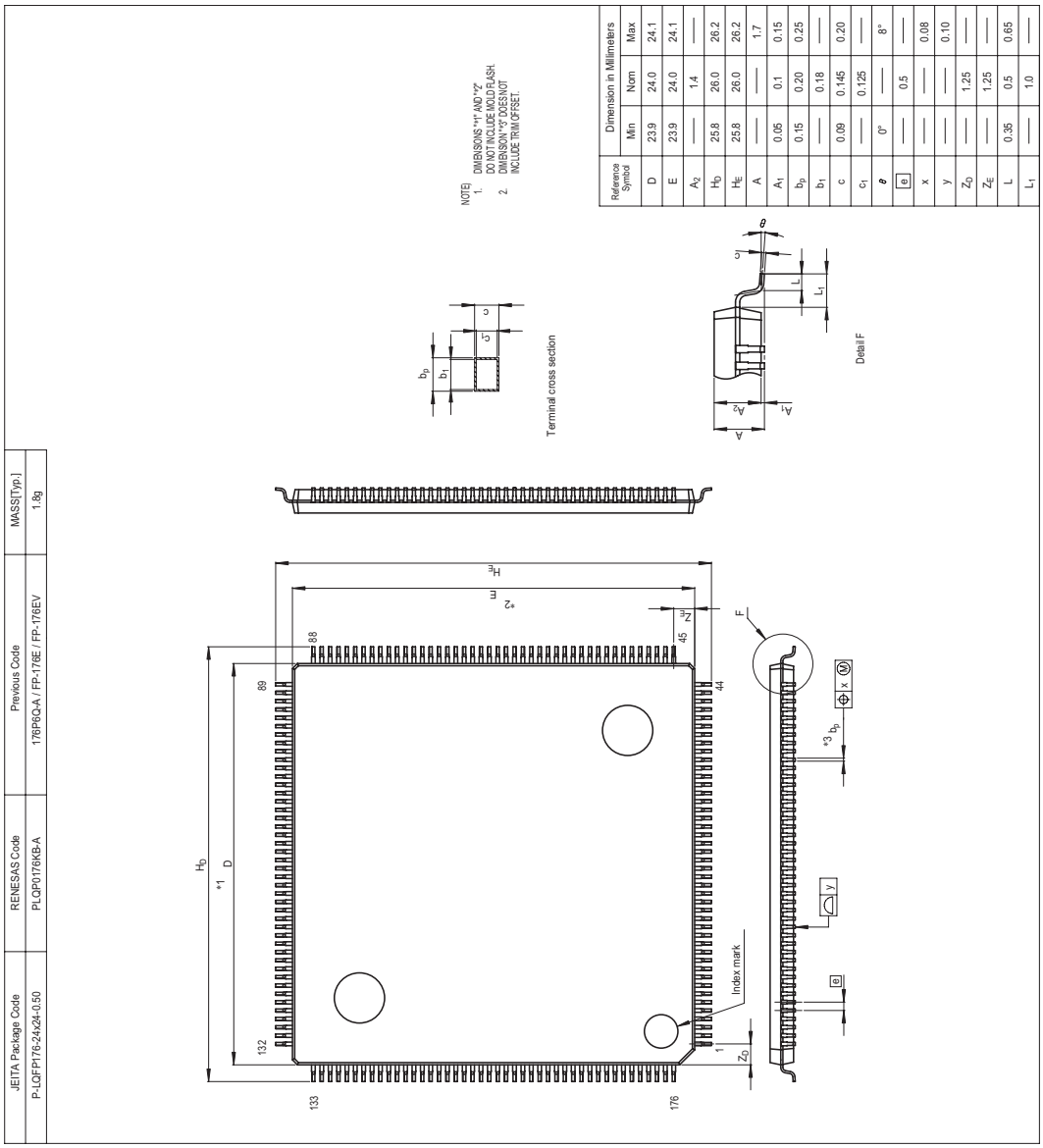
3. IRQ pins that can release deep standby mode are limited to PE7 to PE4 and PC25 to PC22.

4. Z when the TAP controller of the H-UDI is neither the Shift-DR nor Shift-IR state.

5. L when the CKIO output is specified and Z when the CKIO output is stopped with the setting of CKIOCR.

B. Package Dimensions

The package dimension that is shown in the Renesas Semiconductor Package Data Book has priority.



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
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 B.1 Package Dimensions

Main Revisions for This Edition

Item	Page	Revision (See Manual for Details)
All	—	Company name and brand names amended (Before) Renesas Technology Corp. → (After) Renesas Electronics Corporation
2.1.3 System Registers	22	Description amended ... return address from a subroutine procedure. PC points four bytes ahead of the current instruction and controls the flow of the processing.
(3) Program Counter (PC)	23	Description amended PC points four bytes ahead of the instruction being executed.
2.4.2 Data Transfer Instructions Table 2.11 Data Transfer Instructions	46	Table amended

Instruction	Instruction Code	Operation	Execution Cycles	T Bit	Compatibility: SH2, SH2E	SH4	SH-2A
MOVML.L @R15+,Rn	0100nnnn11110101	(R15) → R0, R15 + 4 → R15 (R15) → R1, R15 + 4 → R15 : (R15) → Rn Note: When Rn = R15, read Rn 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 Rn as PR	1 to 16				Yes

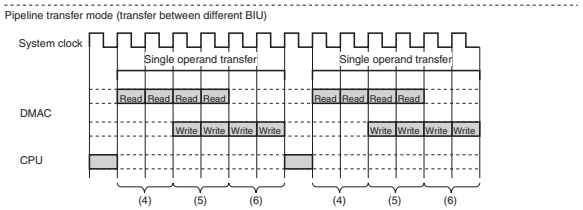
3.1 Features	63	Description deleted • Comprehensive instructions: Single-precision, double-precision, and system control
3.2.2 Non-Numbers (NaN)	66	Description amended • When the EN.V bit in FPSCR is 1, an invalid operation exception will generate FPU exception processing. In this case, the contents of the operation destination register are unchanged.

Item	Page	Revision (See Manual for Details)
3.3.2 Floating-Point Status/Control Register (FPSCR)	69	Table amended Bit / Bit Name / Initial Value / R/W / Description 17 to 12 / Cause / All 0 / R/W / FPU Exception Cause Field 11 to 7 / Enable / All 0 / R/W / FPU Exception Enable Field 6 to 2 / Flag / All 0 / R/W / FPU Exception Flag Field Each time floating-point operation instruction is executed, the FPU exception cause field is cleared to 0 first. When an FPU exception on floating-point operation occurs, the bits corresponding to the FPU exception cause field and FPU exception flag field are set to 1. The FPU exception flag field remains set to 1 until it is cleared to 0 by software. As the bits corresponding to FPU exception enable filed are sets to 1, FPU exception processing occurs. For bit allocations of each field, see table 3.3.
3.5 FPU Exceptions	71	Title amended
3.5.1 FPU Exception Sources	71	Description amended FPU exceptions may occur on floating-point operation instruction and the exception sources are as follows:
3.5.2 FPU Exception Handling	72	Description amended These possibilities of each exceptional handling on floating-point operation are shown in the individual instruction descriptions. All exception events that originate in the floating-point operation are assigned as the same FPU exceptional handling event. The meaning of an exception generated by floating-point operation is determined by software by reading from FPSCR and interpreting the information it contains. Also, the destination register is not changed when FPU exception handling operation occurs.
Section 4 Clock Pulse Generator (CPG)	73 to 76, 78 to 83, 85	Description amended internal clock → CPU clock
4.1 Features (1) PLL Circuit 1	75	Description amended When this is done, the phase of the rising edge of the bus clock is controlled so that it will agree with the phase of the rising edge of the CKIO pin.
4.3 Clock Operating Modes Table 4.3 Relationship between Clock Operating Mode and Frequency Range	82	Description amended Caution: Do not use this LSI for frequency settings other than those in table 4.3.

Item	Page	Revision (See Manual for Details)
5.1.2 Exception Handling Operations Table 5.2 Timing of Exception Source Detection and Start of Exception Handling	93	Table amended **Exception** / **Source** / **Timing of Source Detection and Start of Handling** Instructions / Integer division exceptions / Starts when detecting division-by-zero exception or overflow exception caused by division of the negative maximum value (H'80000000) by −1. FPU exceptions / Exception handling starts triggered by disabled operation exception of floating-point operation instruction (IEEE754 standard), division exception by zero, overflow, underflow, or imprecise exception. Setting the QIS bit in FPSCR or inputting qNaN as well as ±∞ as the floating-point operation instruction source also starts exception handling.
5.2.4 Manual Reset (3) Notes at a Manual Reset	100	Description deleted ... will be deferred until the CPU acquires the bus mastership. The CPU and the BN bit in IBNR of the INTC are initialized by a manual reset.
5.3.1 Address Error Sources Table 5.7 Bus Cycles and Address Errors	101	Table amended **Bus Cycle** / **Type** / **Bus Master** / **Bus Cycle Description** / **Address Errors** Data read/write / CPU / Longword data accessed from other than a long-word boundary / Address error occurs Double longword data accessed from double longword boundary / None (normal) Double longword data accessed from other than double longword boundary / Address error occurs Byte or word data accessed in on-chip peripheral module space*2 / None (normal)
5.3.2 Address Error Exception Handling	102	Note added When an address error occurs, address error exception handling starts after the bus cycle in which the address error occurred ends* and execution of the instruction being executed completes. The CPU operates as follows. ... Note: * In the case of address error related to data read/write. In the case of address error related to instruction fetch, if the bus cycle in which the address error occurred doesn't end until the entire three above-mentioned operations end, the CPU will start address error exception handling again until the bus cycle in which the address error occurred ends.
5.7.1 Types of Exceptions Triggered by Instructions	108	Description amended Exception handling can be triggered by trap instructions, general illegal instructions, slot illegal instructions, integer division exceptions, and FPU exceptions, as shown in table 5.10.

Item	Page	Revision (See Manual for Details)
5.7.1 Types of Exceptions Triggered by Instructions Table 5.10 Types of Exceptions Triggered by Instructions	108	Table amended **Type** / **Source Instruction** / **Comment** FPU exceptions / Instructions that cause disabled operation exception defined by IEEE754 standard or division exception by zero. Instructions that could cause overflow, underflow, or imprecise exception. / FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, FSQRT
5.7.5 Integer Division Exceptions	110	Title and description amended 1. The exception service routine start address which corresponds to the integer division exception that occurred is fetched from the exception handling vector table.
5.7.6 FPU Exceptions	110	Title and description amended An FPU exception handling is generated when the V, Z, O, U or I bit in the FPU exception enable field (Enable) of the floating point status/control register (FPSCR) is set. This indicates the occurrence of an invalid operation exception defined by the IEEE standard 754, a division-by-zero exception, overflow (in the case of an instruction for which this is possible), underflow (in the case of an instruction for which this is possible), or inexact exception (in the case of an instruction for which this is possible). The floating-point operation instructions that may cause generation of an FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT. An FPU exception handling is generated only when the corresponding FPU exception enable bit (Enable) is set. When the FPU detects an exception source by a floating-point operation, FPU operation is halted and FPU exception handling generation is reported to the CPU. When exception handling is started, the CPU operations are as follows. 1. The start address of the exception service routine which corresponding to the FPU exception handling that occurred is fetched from the exception handling vector table.

Item	Page	Revision (See Manual for Details)
5.7.6 FPU Exceptions	111	Description amended The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not an FPU exception handling has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time a floating-point operation instruction is executed. When the V bit in the FPU exception enable field (Enable) of FPSCR is set and the QIS bit in FPSCR is also set, FPU exception handling is generated when qNAN or ±∞ is input to a floating point operation instruction source.
5.9 Stack Status after Exception Handling Ends Table 5.12 Stack Status After Exception Handling Ends	113	Table amended Exception Type: Integer division exception Stack Status: SP → Start address of relevant integer division instruction (32 bits); SR (32 bits)
6.5 Interrupt Exception Handling Vector Table and Priority Table 6.4 Interrupt Exception Handling Vectors and Priorities	137 to 145	Table amended Interrupt Source; Interrupt Vector: Vector, Vector Table Address Offset; Interrupt Priority (Initial Value); Corresponding IPR (Bit); IPR Setting Unit Internal Priority; Default Priority
6.8 Register Banks Figure 6.10 Overview of Register Bank Configuration	154	Figure amended Register banks; Bank 0; Bank 1; ....; Bank 14 R0; R1; ⋮; R14; GBR; MACH; MACL; PR; VTO Interrupt generated (save) RESBANK instruction (restore)

Item	Page	Revision (See Manual for Details)
8.4.4 Notes	198	Description amended 1. Programs that access memory-mapped cache of the operand cache should be placed in a cache-disabled space. Programs that access memory-mapped cache of the instruction cache should be placed in a cache-disabled space, and in each of the beginning and the end of that, two or more read accesses to on-chip peripheral modules or external address space (cache-disabled address) should be executed.
9.4.8 SDRAM Refresh Control Register 1 (SDRFCNT1)	223	Description amended DRFC = (Auto-refresh request interval / Bus clock cycle) – 1
11.4.1 DMA Transfer Mode Figure 11.2 Examples of the Alternation of Bus Mastership between the DMAC and CPU in Various DMA Transfer Modes	341	Figure amended  However, when a DMA access to external address space followed by CPU access to external address space is occurred, CPU access next to DMA cycle may not be occurred.
11.4.2 DMA Transfer Condition (1) Unit Operand Transfer	342	Description added ... transfer is completed by repeating unit transfer operations until the byte counter does reach 0. In the case that the DMA transfer condition is the unit operand transfer and the input sense mode of DMA request is the level sense, there is the mask period of the DMA request in the channel arbitration period after one operand transfer end (please refer to section 11.7.3, Sense Mode for DMA Requests for details). Therefore, in the channel arbitration period after one operand transfer end, in the case that there is no DMA request of the higher-priority channel than the transferring channel and there is the DMA request of the lower-priority channel than the transferring channel, the DMA transfer of the low-priority channel starts. To execute the DMA transfer of the high-priority channel in succession, please set the DMA transfer condition to the sequential operand transfer or the non-stop transfer.

Item	Page	Revision (See Manual for Details)
11.4.2 DMA Transfer Condition (2) Sequential Operand Transfer	342	Description added ... unless there is a DMA request from a higher-priority channel. In the case that the DMA transfer condition is the sequential operand transfer, even if the input sense mode of DMA request is the level sense, there is no mask period before the byte count becomes 0. Therefore, the DMA transfer of the low-priority channel than the transferring channel cannot start.
12.1 Features Table 12.1 MTU2 Functions	369	Table amended **Item** / **Channel 0** / **Channel 1** / **Channel 2** / **Channel 3** / **Channel 4** / **Channel 5** DMAC activation: TGR compare match or input capture (Channel 0) / TGR compare match or input capture (Channel 1) / TGR compare match or input capture (Channel 2) / TGR compare match or input capture (Channel 3) / TGR compare match or input capture and TCNT overflow or underflow (Channel 4) / — (Channel 5)
12.3.31 Timer Waveform Control Register (TWCR)	453	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 0 / WRE / 0 / R/(W) / Initial Output Suppression Enable Selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is suppressed only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value is also output when synchronous clearing occurs in the Tb interval at the trough immediately after TCNT_3 and TCNT_4 start operation. For the Tb interval at the trough in complementary PWM mode, see figure 12.40. 0: Outputs the initial value specified in TOCR 1: Suppresses initial output [Setting condition] • When 1 is written to WRE after reading WRE = 0

Item	Page	Revision (See Manual for Details)
12.4.8 Complementary PWM Mode (2) Outline of Complementary PWM Mode Operation (n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode	509	Description added ... suppressed. When using the initial output suppression function, make sure to set compare registers TGRB_3, TGRA_4, and TGRB_4 to a value twice or more the setting of dead time data register TDDR. If synchronous clearing occurs with the compare registers set to a value less than twice the setting of TDDR, the PWM output dead time may be too short (or nonexistent) or illegal active-level PWM negative-phase output may occur during the initial output suppression interval. For details, see section 12.7.23, Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode.
(3) Interrupt Skipping in Complementary PWM Mode: (c) Buffer Transfer Control Linked with Interrupt Skipping Figure 12.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0)	522	Figure replaced
Figure 12.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period	523	Figure replaced
12.5.3 A/D Converter Activation (3) A/D Converter Activation by A/D Converter Start Request Delaying Function	535	Description amended The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the TCNT_4 count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1.
12.7.23 Notes on Output Waveform Control During Synchronous Counter Clearing in Complementary PWM Mode	564, 565	Newly added
14.5.3 Interval Timer Overflow Flag	636	Newly added

Item	Page	Revision (See Manual for Details)
14.5.5 Manual Reset in Watchdog Timer Mode	637	Description deleted ... CPU acquires the bus mastership.
15.5.3 Transition to Standby Mode after Setting Register	665	Description amended ... mode after waiting for two count clocks or more.
15.5.4 Crystal Oscillator Circuit for RTC Figure 15.6 Example of Connecting Crystal Oscillator Circuit for RTC	666	Figure amended Notes: 7. When not using a crystal oscillation circuit for RTC, fix the RTC_X1 pin (pull-up, pull-down, connect to power supply, or connect to ground) and leave the RTC_X2 pin open.
16.3.6 Serial Control Register (SCSCR)	682	Table amended

Bit	Bit Name	Initial Value	R/W	Description
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.

16.3.8 Bit Rate Register (SCBRR) — 694 — Table amended

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (2)

	Pφ (MHz)											
	8			9.8304			10			12		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34

Table 16.4 Bit Rates and SCBRR Settings (Asynchronous Mode) (5) — 695 — Table amended

	Pφ (MHz)														
	30			33			36			38			40		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
4800	0	194	0.16	0	214	–0.07	0	233	0.16	0	246	0.16	1	64	0.16

Table 16.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (1) — 696 — Table amended

	Pφ (MHz)									
Bit Rate	5		8		16		28.7		30	
(bit/s)	n	N	n	N	n	N	n	N	n	N
250	3	77	3	124	3	249				
1 M	—	—			0	3	—	—	—	—
2 M							—	—	—	—

Item	Page	Revision (See Manual for Details)
Table 16.5 Bit Rates and SCBRR Settings (Clocked Synchronous Mode) (2)	697	Table amended

Bit Rate (bit/s)	Pϕ (MHz) 33 n	33 N	36 n	36 N	38 n	38 N	40 n	40 N
250								
500	3	255	—	—				
1 k	3	128	3	140	3	147	3	155
2.5 k	2	205	2	224	2	237	2	249
5 k	2	102	2	112	2	118	2	124
10 k	1	205	1	224	1	237	1	249
25 k	1	82	1	89	1	94	1	99
50 k	0	164	0	179	0	189	0	199
100 k	0	82	0	89	0	94	0	99
250 k	0	32	0	35	0	37	0	39
500 k	—	—	0	17	0	18	0	19
1 M	—	—	0	8	—	—	0	9
2 M	—	—	—	—	—	—	0	4

Note and description amended

[Legend]

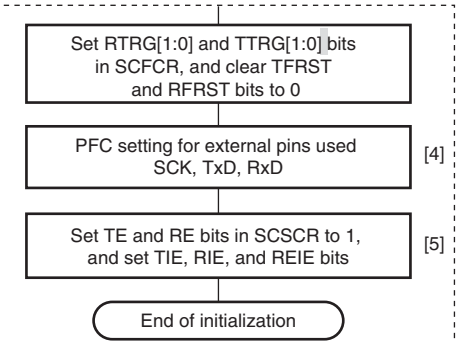
Blank: No setting possible, or it is not possible to satisfy the electrical characteristics of the MCU regardless of the communication partner device.

—: Setting possible, but error occurs

Table 16.6 indicates the maximum bit rates in asynchronous mode when the baud rate generator is used. Tables 16.7 and 16.8 list the maximum rates when the external clock input is used (when tscyc = 12 tpcyc*).

Note: * Make sure that the electrical characteristics of this MCU and that of a connected MCU are satisfied.

16.4.2 Operation in Asynchronous Mode Figure 16.3 Sample Flowchart for SCIF Initialization	711	Figure amended



Item | **Page** | **Revision (See Manual for Details)**

16.5 SCIF Interrupts — 725

Description amended

When the RIE bit is set to 0 and the REIE bit is set to 1, the SCIF requests only an ERI or a BRI interrupt without requesting an RXI interrupt.

17.3.1 I^2C Bus Control Register 1 (ICCR1) — 733

Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I^2C Bus Interface 3 Enable 0: SCL and SDA output is disabled. (Input to SCL and SDA is enabled.) 1: This bit is enabled for transfer operations.

17.3.2 I^2C Bus Control Register 2 (ICCR2) — 737

Table amended

Bit	Bit Name	Value	R/W	Description
1	IICRST	0	R/W	IIC Control Part Reset Resets bits BC[2:0] in ICMR and internal circuits. If this bit is set to 1 when hang-up occurs because of communication failure during I^2C bus operation, bits BC[2:0] in ICMR and internal circuits can be reset.

17.3.3 I^2C Bus Mode Register (ICMR) — 739

Table amended

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	BC[2:0]	000	R/W	Bit Counter These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. With the I^2C bus format, the data is transferred with one addition acknowledge bit. Should be made between transfer frames. If these bits are set to a value other than B'000, the setting should be made while the SCL pin is low. The bit value returns to B'000 automatically at the end of a data transfer including the acknowledge bit. And the value becomes B'111 automatically after the stop condition detection. These bits are cleared by a power-on reset, in deep standby mode, software standby mode, or 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.

17.3.4 I^2C Bus Interrupt Enable Register (ICIER) — 740

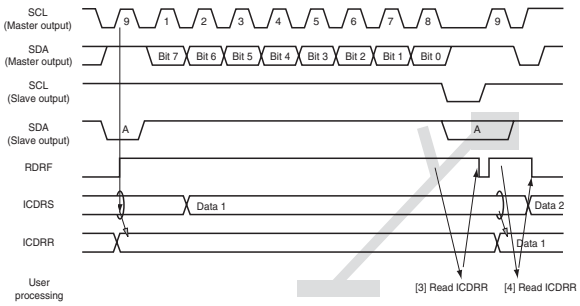
Table amended

Bit	Bit Name	Initial Value	R/W	Description
5	RIE	0	R/W	Receive Interrupt Enable Enables or disables the receive data full interrupt request (RXI) when receive data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Receive data full interrupt request (RXI) are disabled. 1: Receive data full interrupt request (RXI) are enabled.

741

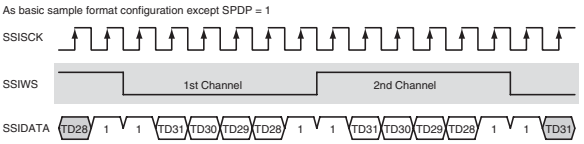
Table amended

Bit	Bit Name	Initial Value	R/W	Description
4	NAKIE	0	R/W	NACK Receive Interrupt Enable Enables or disables the NACK detection, arbitration lost and overrun error interrupt request (NAKI) when the NACKF or AL/OVE bit in ICSR is set. NAKI can be canceled by clearing the NACKF, AL/OVE, or NAKIE bit to 0. 0: NACK receive interrupt request (NAKI) is disabled. 1: NACK receive interrupt request (NAKI) is enabled.

Item	Page	Revision (See Manual for Details)
17.4.5 Slave Receive Operation Figure 17.12 Slave Receive Mode Operation Timing (2)	757	Figure amended 
17.6 Bit Synchronous Circuit Table 17.5 Time for Monitoring SCL	769	Table amended CKS3 / CKS2 / Time for Monitoring SCL*1 0 / 0 / 9 tpcyc*2 0 / 1 / 21 tpcyc*2 1 / 0 / 39 tpcyc*2 1 / 1 / 87 tpcyc*2
17.7.1 Issuance of Stop Condition and Start Condition (Retransmission)	770	Description deleted ... may not be output correctly.
17.7.2 Note on Setting for Multi-Master Operation	770	Description replaced
17.7.3 Reading ICDRR in Master Receive Mode	—	Description deleted
17.7.3 Note on Master Receive Mode	770	Newly added
17.7.4 Note on Setting ACKBT in Master Receive Mode	770	Newly added
17.7.5 Note on the States of Bits MST and TRN when Arbitration is Lost	771	Newly added
17.7.6 Note on IICRST and BBSY bits	771	Newly added
Section 18 Serial Sound Interface (SSI)	773	Description amended The serial sound interface (hereinafter referred to as the "SSI") is a transceiver module designed to send or receive audio data interface with a variety of devices compatible with I^2S bus.

Item	Page	Revision (See Manual for Details)
18.2 Input/Output Pins Table 18.1 Pin Assignments	775	Table amended **Pin Name** / **Number of Pins** / **I/O** / **Description** AUDIO_CLK / 1 / Input / External clock for audio (Oversample clock) AUDIO_X1 / 1 / Input / Crystal oscillator for audio (Oversample clock) AUDIO_X1 / 1 / Output /
18.3.1 Control Register (SSICR)	778	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 15 / SCKD / 0 / R/W / Serial Bit Clock Direction 0: Serial bit clock is input, slave mode. 1: Serial bit clock is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
	779	Table amended **Bit** / **Bit Name** / **Initial Value** / **R/W** / **Description** 14 / SWSD / 0 / R/W / Serial WS Direction 0: Serial word select is input, slave mode. 1: Serial word select is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) and (1,1). Other settings are prohibited.
18.4.1 Bus Format Table 18.3 Bus Format for SSI Module	789	Table amended / **Non-Compressed Slave Receiver** / **Non-Compressed Slave Transmitter** / **Non-Compressed Master Receiver** / **Non-Compressed Master Transmitter** TRMD / 0 / 1 / 0 / 1 SCKD / 0 / 0 / 1 / 1
18.4.2 Non-Compressed Modes	790	Description amended The non-compressed modes support all serial audio streams split into channels. It supports I^2S compatible format as well as many more variants on these modes.
(3) Master Receiver	790	Description amended ... select signals are internally derived from the oversampling clock.
(4) Master Transmitter		Description amended ... signals are internally derived from the oversampling clock.
(5) Operating Setting Related to Word Length		Description amended All bits related to the SSICR's word length are valid in non-compressed modes. The SSI module supports many configurations, but the formats described below are I^2S compatible, MSB-first left-aligned, and MSB-first right-aligned.

Item	Page	Revision (See Manual for Details)
18.4.2 Non-Compressed Modes (5) Operating Setting Related to Word Length	791	Description amended 1. I^2S Compatible Format Figures 18.3 and 18.4 demonstrate the supported I^2S compatible format both with and without padding. Padding occurs when the data word length is smaller than the system word length.
Figure 18.3 I^2S Compatible Format (without Padding)		Figure title amended
Figure 18.4 I^2S Compatible Format (with Padding)		Figure title amended
	792	Description amended Figure 18.5 shows MSB-first left-aligned format, and figure 18.6 shows MSB-first right-aligned format. 2. MSB-First Left-Aligned Format
Figure 18.5 MSB-First Left-Aligned Format (Transmitted and Received in the order of Serial Data and Padding Bits)		Figure title amended
		Description amended 3. MSB-First Right-Aligned Format
Figure 18.6 MSB-First Right-Aligned Format (Transmitted and Received in the order of Padding Bits and Serial Data)		Figure title amended
(6) Multi-channel Formats	793	Description amended Some devices extend the definition of the specification by I^2S bus and allow more than 2 channels to be transferred within two system words. The SSI module supports the transfer of 4, 6 and 8 channels by using the CHNL, SWL and DWL bits only when the system word length (SWL) is greater than or equal to the data word length (DWL) multiplied by channels (CHNL).
	795	Description amended Figures 18.7 to 18.9 show how 4, 6 and 8 channels are transferred to the serial audio bus.
Figure 18.7 Multichannel Format (4 Channels Without Padding)		Figure title amended

Item	Page	Revision (See Manual for Details)
18.4.2 Non-Compressed Modes (6) Multi-channel Formats Figure 18.8 Multichannel Format (6 Channels with High Padding)	795	Figure title amended
Figure 18.9 Multichannel Format (8 Channels; Transmitting and Receiving in the order of Padding Bits and Serial Data; with Padding)	796	Figure title amended
(7) Bit Setting Configuration Format Figure 18.13 Inverted Padding Polarity	797	Figure amended 
18.4.4 Transmit Operation	801	Note amended Note: * Input clock from the SSISCK pin when SCKD = 0. Oversampling clock when SCKD = 1.
18.4.5 Receive Operation	804	Note amended Note: * Input clock from the SSISCK pin when SCKD = 0. Oversampling clock when SCKD = 1.
18.4.7 Serial Bit Clock Control	808	Description amended If the serial clock direction is set to output (SCKD = 1), this module is in clock master mode, and the shift register uses the oversampling clock or a divided oversampling clock as the bit clock. The oversampling clock is divided by the ratio specified by the serial oversampling clock division ratio bits (CKDV) in SSICR for use as the bit clock by the shift register.
18.5.1 Limitations from Overflow during Receive DMA Operation	809	Description amended ... Therefore, data to be received at the L channel may sometimes be received at the R channel if an overflow occurs, for example, under the following condition: the control register (SSICR) has a 32-bit setting for both data word length (DWL2 to DWL0) and system word length (SWL2 to SWL0).

Item	Page	Revision (See Manual for Details)
19.2.1 Block Diagram	814	Description amended **Important:** Although core of RCAN-ET is designed based on a 32-bit bus system, the whole RCAN-ET including MPI for the CPU has 16-bit bus interface to CPU. LongWord (32-bit) accesses are converted into two consecutive word accesses by the bus interface.
19.4.3 Bit Configuration Register (BCR0, BCR1)	834	Description amended Where: BRP (Baud Rate Pre-scaler) is the value stored in BCR0 incremented by 1 and fclk is the used peripheral clock frequency.
• BCR0 (Address = H'006)	836	Description amended Bits 7 to 0—Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0]): These bits are used to define the peripheral clock periods contained in a Time Quantum.

Table amended

Bit 7: BRP[7]	Bit 6: BRP[6]	Bit 5: BRP[5]	Bit 4: BRP[4]	Bit 3: BRP[3]	Bit 2: BRP[2]	Bit 1: BRP[1]	Bit 0: BRP[0]	Description
0	0	0	0	0	0	0	0	2 × peripheral clock (Initial value)
0	0	0	0	0	0	0	1	4 × peripheral clock
0	0	0	0	0	0	1	0	6 × peripheral clock
:	:	:	:	:	:	:	:	2 × (register value+1) × peripheral clock
1	1	1	1	1	1	1	1	512 × peripheral clock

Item	Page	Revision (See Manual for Details)
19.4.4 Interrupt Request Register (IRR)	839	Description amended The interrupt request register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.
19.6.1 Configuration of RCAN-ET Figure 19.6 Reset Sequence	859	Figure amended Notes: 3. It takes approximately one bit time for GSR[3] to be cleared to 0.
20.3.2 A/D Control/Status Register (ADCSR)	885	Table amended

Bit	Bit Name	Initial Value	R/W	Description
2 to 0	CH[2:0]	000	R/W	Channel Select These bits and the MDS bits in ADCSR select the analog input channels.

MDS = 0xx	MDS = 100 or MDS = 110	MDS = 101 or MDS = 111
000: AN0	000: AN0	000: AN0
001: AN1	001: AN0, AN1	001: AN0, AN1
010: AN2	010: AN0 to AN2	010: AN0 to AN2
011: AN3	011: AN0 to AN3	011: AN0 to AN3
100: AN4	100: AN4	100: AN0 to AN4
101: AN5	101: AN4, AN5	101: AN0 to AN5
110: AN6	110: AN4 to AN6	110: AN0 to AN6
111: AN7	111: AN4 to AN7	111: AN0 to AN7

Item	Page	Revision (See Manual for Details)
20.3.2 A/D Control/Status Register (ADCSR)	885	Note added Notes: 1. Only 0 can be written to clear the flag after 1 is read. Please note that ADF flag becomes "0" in the following cases, too. (1) Reading the state of ADF = 1 with CPU. (2) Clearing ADF flag by having read ADDR with DMAC (3) Set of ADF flag according to A/D conversion end (4) Writing 0 in the ADF flag with CPU
20.5 Interrupt Sources and DMAC Transfer Request	897	Description amended ... set to 1 on completion of A/D conversion. Note that the direct memory access controller (DMAC) can be activated by an ADI interrupt depending on the interrupt controller (INTC) setting. ... of data. To make the DMAC transfer all conversion data, set the ADDR where A/D conversion data is stored as the transfer source address, the number of converted channels × 2 as the transfer byte count, and continuous operand transfer or non-stop transfer as the DMA transfer condition.
20.7.7 Note on Usage in Scan Mode and Multi Mode	902	Description replaced
26.5 Usage Notes	1026	Description amended 4. If the $\overline{\text{UDTRST}}$ pin is asserted immediately after the setting of the UDTDO transition timing switching command and the negation of the RES pin, the UDTDO transition timing switching command is cleared. To avoid this case, make sure to put 20 tcyc or longer between the signal transition timing of the $\overline{\text{RES}}$ and $\overline{\text{UDTRST}}$ pins. For details, see section 26.4.3, UDTDO Output Timing.
28.2 Register Bits	1078	Table amended

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFCR_0	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	SCIF
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	

Item	Page	Revision (See Manual for Details)
28.2 Register Bits	1079	Table amended

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFCR_1	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	SCIF
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFCR_2	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	

	1080	Table amended

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFCR_3	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	SCIF
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFCR_4	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFCR_5	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	

	1081	Table amended

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
SCFCR_6	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	SCIF
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	
SCFCR_7	—	—	—	—	—	RSTRG2	RSTRG1	RSTRG0	
	RTRG1	RTRG0	TTRG1	TTRG0	—	TFRST	RFRST	LOOP	

	1083	Table amended

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
BCR1_0	TSG1_3	TSG1_2	TSG1_1	TSG1_0	—	TSG2_2	TSG2_1	TSG2_0	RCAN-ET
	—	—	SJW1	SJW0	—	—	—	BSP	

	1086	Table amended

Register Abbreviation	Bits 31/ 23/15/7	Bits30/ 22/14/6	Bits 29/ 21/13/5	Bits28/ 20/12/4	Bits 27/ 19/11/3	Bits26/ 18/10/2	Bits 25/ 17/9/1	Bits24/ 16/8/0	Module
BCR1_1	TSG1_3	TSG1_2	TSG1_1	TSG1_0	—	TSG2_2	TSG2_1	TSG2_0	RCAN-ET
	—	—	SJW1	SJW0	—	—	—	BSP	

28.3 Register States in Each Operating Mode	1095	Table amended

Register Abbreviation	Power-on Reset	Manual Reset	Software Standby	Deep Standby	Module Standby	Sleep	Module
RSECAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	RTC
RMINAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RHRAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RWKAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RDAYAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RMONAR	Retained*8	Retained	Retained	Retained*8	Retained	Retained	
RCR1	Initialized	Initialized	Retained	Initialized	Retained	Retained	
RCR2	Initialized	Initialized*6	Retained	Initialized	Retained	Retained	
RYRAR	Retained	Retained	Retained	Retained	Retained	Retained	
RCR3	Initialized	Retained	Retained	Initialized	Retained	Retained	

	1106	Note added
		Notes: 8. The ENB bit is initialized.
29.3 AC Characteristics	1116	Table amended

Table 29.4 Maximum Operating Frequency

Item		Symbol	Min.	Typ.	Max.	Unit	Remarks
Operating frequency	CPU clock (Iϕ)	f	20	—	120	MHz	Regular specifications
					100		Wide-range specifications
	Bus clock (Bϕ)		20	—	60		
	Peripheral clock (Pϕ)		5	—	40		

Item | **Page** | **Revision (See Manual for Details)**

29.3.1 Clock Timing
Table 29.5 Clock Timing

1117 — Table amended

Item	Symbol	Min.	Max.	Unit	Figure
CKIO clock input frequency	f_{CK}	20	60	MHz	Figure 29.2
CKIO clock input cycle time	t_{CKIcyc}	16.67	50	ns	
CKIO clock input pulse low width	t_{CKIL}	0.4	0.6	t_{CKIcyc}	
CKIO clock input pulse high width	t_{CKIH}	0.4	0.6	t_{CKIcyc}	
CKIO clock input rise time	t_{CKIr}	—	3	ns	
CKIO clock input fall time	t_{CKIf}	—	3	ns	
CKIO clock output frequency	f_{OP}	20	60	MHz	Figure 29.3
CKIO clock output cycle time	t_{cyc}	16.67	50	ns	
CKIO clock output pulse low width	t_{CKOL}	$t_{cyc}/2 - t_{CKOr}$	—	ns	
CKIO clock output pulse high width	t_{CKOH}	$t_{cyc}/2 - t_{CKOf}$	—	ns	
CKIO clock output rise time	t_{CKOr}	—	3	ns	
CKIO clock output fall time	t_{CKOf}	—	3	ns	

29.3.3 Bus Timing
Table 29.7 Bus Timing

1123 — Note amended

Notes: 2. The maximum value (f_{max}) of Bϕ (bus clock) depends on the number of wait cycles and the system configuration of your board.

Figure 29.10 (1) External Address Space: Basic Bus Timing (Normal Access, Read/Write Cycle Wait = 3, CS Assert Wait = 1, Write Data Output Wait = 1, WR/RD Assert Wait = 2, Write Data Output Delay Cycles = 0, Read/Write CS Delay Cycles = 1)

1123 — Figure and figure title amended

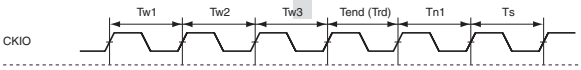
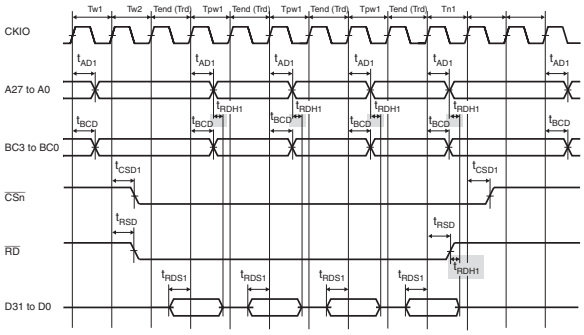
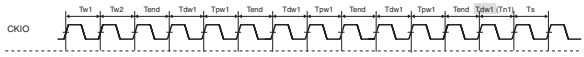


Figure 29.10 (2) External Address Space: Basic Bus Timing (Normal Access, Data Recovery Cycles = 0, Read/Write Cycle Wait = 1, Read/Write CS Delay Cycles = 1, Other Wait Settings = 0)

1124 — Figure added

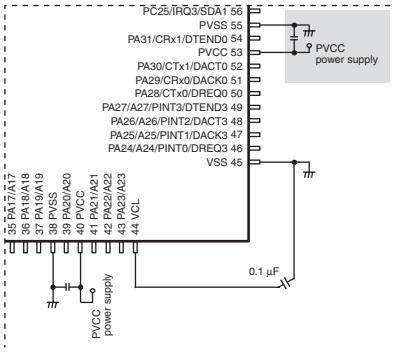
Figure 29.10 (3) External Address Space: Basic Bus Timing (Normal Access, Data Recovery Cycles = 2, Read/Write Cycle Wait = 1, Read/Write CS Delay Cycles = 1, Other Wait Settings = 0)

1125 — Figure added

Item	Page	Revision (See Manual for Details)
29.3.3 Bus Timing Figure 29.11 External Address Space: Basic Bus Timing (Page Read Access, Normal Access Compatible Mode , Read Cycle Wait = 2, Page Read Cycle Wait = 2, CS Assert Wait = 1, RD Assert Wait = 1, Read CS Delay Cycles = 1)	1126	Figure title amended
Figure 29.12 External Address Space: Basic Bus Timing (Page Read Access, External Read Data Continuous Assert Mode, Read Cycle Wait = 2, Page Read Cycle Wait = 1, CS Assert Wait = 1, RD Assert Wait = 1, Read CS Delay Cycles = 1)	1126	Figure and figure title amended 
Figure 29.13 External Address Space: Basic Bus Timing (Page Write Access, Write Cycle Wait = 2, CS Assert Wait = 1, WR Assert Wait = 1, Write Data Output Delay Cycles = 1, Other Wait Settings = 0)	1127	Figure and figure title amended 
Figure 29.14 External Address Space: Timing with External Wait (Page Read Access to 16-Bit Width Channel, External Read Data Continuous Assert Mode, Read Cycle Wait = 3, Page Read Cycle Wait = 3, Other Wait Settings = 0, External Wait Cycles = 2)	1128	Figure and figure title amended BC3 to BC0 → BC1, BC0 D31 to D0 → D15 to D0
Figure 29.15 Single Read Bus Timing for SDRAM Space (DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))	1129	Figure title amended

Item	Page	Revision (See Manual for Details)
29.3.3 Bus Timing Figure 29.16 Single Write Bus Timing for SDRAM Space (DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))	1130	Figure title amended
Figure 29.17 Multiple Read Bus Timing for SDRAM Space (Four Data Access, DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))	1131	Figure title amended
Figure 29.18 Multiple Write Bus Timing for SDRAM Space (Four Data Access, DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))	1132	Figure title amended
Figure 29.19 Multiple Read Row Span Bus Timing for SDRAM Space (Eight Data Access, DCL = 2 (Two Cycles), DRCD = 1 (Two Cycles), DPCG = 1 (Two Cycles))	1133	Figure title amended
29.3.6 MTU2 Module Timing Table 29.10 MTU2 Module Timing	1138	Table amended Note amended Note: t_{pcyc} indicates peripheral clock (Pϕ) cycle.
29.6 Usage Note Figure 29.49 Example of Externally Allocated Capacitors	1153	Figure amended

Item	Symbol	Min.	Max.	Unit	Figure
Output compare output delay time	t_{TOCD}	—	100	ns	Figure 31.25
Input capture input setup time	t_{TICS}	20	—	ns	
Timer input setup time	t_{TCKS}	20	—	ns	Figure 31.26



Index

Numerics

16-bit counter mode 617
16-bit/32-bit displacement 29
8-bit timers (TMR) 599

A

A/D conversion time
(multi mode and scan mode)..... 896
A/D conversion time (single mode)..... 895
A/D conversion timing 895
A/D converter (ADC) 877
A/D converter activation..... 534
A/D converter characteristics..... 1151
A/D converter start request
delaying function 524
A/D trigger input timing 1146
Absolute address..... 29
Absolute address accessing..... 29
Absolute maximum ratings..... 1107
AC characteristics..... 1116
AC characteristics measurement
conditions..... 1150
Address array..... 182, 194
Address array read 194
Address errors..... 101
Address map 203
Address map for each mailbox 817
Address-array write
(associative operation)..... 195
Address-array write
(non-associative operation)..... 195
Addressing modes..... 30
Advanced User Debugger II (AUD-II) . 1027
Analog input pin ratings 901
Arithmetic operation instructions 48

B

Bit manipulation instructions 59
Bit synchronous circuit 767
Block diagram..... 9
Branch instructions 53
Break detection and processing..... 727
Break on data access cycle..... 175
Break on instruction fetch cycle..... 174
Bus state controller (BSC) 199
Bus timing..... 1122

C

Cache 181
Calculating exception handling
vector table addresses..... 96
CAN interface 815
CAN sleep mode 861
Canceling software standby mode 633
Cascaded connection..... 617
Changing the division ratio 88
Changing the frequency 87, 633
Changing the multiplication rate..... 87
Clock frequency control circuit..... 75
Clock operating modes..... 77
Clock pulse generator (CPG) 73
Clock timing 1116
Clocked synchronous serial format..... 757
Coherency of cache and
external memory 194
Compare match count mode 617
Compare match signal..... 614
Complementary PWM mode 487
Control signal timing 1120
Controller area network (RCAN-ET)..... 811
CPU..... 19
Crystal oscillator 75

CSC interface 237

D

D/A converter (DAC) 903
D/A converter characteristics 1152
D/A output hold function in software standby mode 908
Data array 182, 195
Data array read 196
Data array write 196
Data format in registers 24
Data formats in memory 24
Data transfer instructions 44
DC characteristics 1108
Dead time compensation 529
Definitions of A/D conversion accuracy 897
Delayed branch instructions 27
Denormalized numbers 66
Direct memory access controller (DMAC) 299
Displacement accessing 29
Divider 75

E

Effective address calculation 30
Electrical characteristics 1107
Equation for getting SCBRR value 692
Exception handling 91
Exception handling state 61
Exception handling vector table 95
Exception source generation immediately after delayed branch instruction 111
Exceptions triggered by instructions 108
External pulse width measurement 528
External trigger input timing 896

F

Floating point operation instructions 56
Floating-point ranges 65
Floating-point registers 67
Format of double-precision floating-point number 63
Format of single-precision floating-point number 63
FPU exception 71
FPU-related CPU instructions 58
Full-scale error 898

G

General illegal instructions 109
General registers 19
Global base register (GBR) 22

H

Halt mode 860
H-UDI commands 1022
H-UDI interrupt 133, 1025
H-UDI reset 1025
H-UDI-related pin timing 1147

I

I/O port timing 1146
I/O ports 909
I^2C bus format 748
I^2C bus interface 3 (IIC3) 729
I^2C bus interface 3 (IIC3) usage note 770
ID Reorder 826
IIC3 module timing 1142
Immediate data 28
Immediate data accessing 28
Immediate data format 25
Initial values of general registers 23
Instruction features 26

Instruction format 34
Instruction set 38
Integer division instructions..... 110
Interrupt exception handling..... 107
Interrupt exception handling vectors and priorities 137
Interrupt priority level..... 106
Interrupt response time 149
IRQ interrupts 134

J

Jump table base register (TBR) 22

L

List of registers 1033
Load-store architecture 27
Local acceptance filter mask (LAFM).... 823
Logic operation instructions 51
LRU 183

M

Mailbox..... 814
Mailbox control 815
Mailbox structure..... 818
Manual reset 100
Master receive operation..... 751
Master transmit operation 749
Measurement circuit 1150
Memory-mapped cache 194
Message control field..... 819
Message data fields..... 824
Message receive sequence 867
Message transmission sequence..... 865
Micro processor interface (MPI)..... 814
Module standby function 1017
MTU2 functions 368
MTU2 interrupts 532
MTU2 module timing 1138
MTU2 output pin initialization 566
Multi mode..... 889
Multi-function timer pulse unit 2 (MTU2)..... 367
Multiplexed pin table (Port A)..... 927
Multiplexed pin table (Port B)..... 929
Multiplexed pin table (Port C)..... 931
Multiplexed pin table (Port D)..... 933
Multiplexed pin table (Port E)..... 934
Multiplexed pin table (Port F)..... 934
Multiply and accumulate register high (MACH)..... 22
Multiply and accumulate register low (MACL) 22
Multiply/Multiply-and-accumulate operations..... 27

N

NaN..... 66
NMI interrupt..... 133
Noise filter 761
Nonlinearity error..... 898
Non-numbers (NaN) 65
Note on making a transition to deep standby mode..... 1016
Note on using a PLL oscillation circuit..... 90
Note on using crystal resonator..... 89

O

Offset error..... 898
On-chip peripheral module interrupts 135
On-chip RAM 989
Operation in asynchronous mode..... 708
Operation in clocked synchronous mode..... 717

P

Package dimensions 1160
Page conflict 990
Permissible signal source impedance 901
Pin assignments 10
Pin function controller (PFC) 927
PINT interrupts 135
PLL circuit 75
Power-down modes 991
Power-down state 61
Power-on reset 98
Prefetch operation
(only for operand cache) 191
Procedure register (PR) 23
Processing of analog input pins 900
Program counter (PC) 23
Program execution state 61

Q

Quantization error 898

R

RCAN-ET bit rate calculation 837
RCAN-ET interrupt sources 871
RCAN-ET memory map 816
Realtime clock (RTC) 639
Receive data sampling timing and
receive margin (asynchronous mode) 728
Reconfiguration of mailbox 869
Register bank error
exception handling 104, 158
Register bank errors 104
Register bank exception 158
Register banks 23, 154
Register banks and bank
control registers 155
Registers
ABACK0 853
ACSWR 236
ADCSR 882
ADDR 880
BAMR 165
BAR 164
BBR 168
BCR 834
BDMR 167
BDR 166
BRCR 170
CCR 184
CKIOCR 86
CS1WCNTn 215
CS2WCNTn 217
CSMODn 212
CSnCNT 207
CSnREC 209
DACR 905
DADR0 905
DADR1 905
DMASTS 338
DMCBCT 308
DMCDADR 307
DMCNTA 318
DMCNTB 326
DMCSADR 306
DMEDET 336
DMICNT 333
DMICNTA 334
DMISTS 335
DMMOD 312
DMRBCT 311
DMRDADR 310
DMRSADR 309
DMSCNT 332
DREQER 129
DSCNT 1005
DSFR 1006
FPSCR 68
FPUL 69

FRQCR 83
GSR 831
IBCR 127
IBNR 128
ICCR 733
ICDRR 746
ICDRS 746
ICDRT 746
ICIER 740
ICMR 738
ICR 121
ICSR 742
IMR 845
IPR 119
IRQRR 123
IRR 839
MBIMR0 856
MCR 825
NF2CYC 747
PACR 938
PADR 910
PAIOR 937
PAPR 912
PBCR 949
PBDR 914
PBIOR 948
PBPR 916
PCCR 963
PCDR 918
PCIOR 962
PCPR 919
PDCR 974
PDDR 921
PDIOR 973
PDPR 922
PECR 981
PEPR 923
PFCR 984
PFDR 925
PFIOR 983
PFPR 926
PINTER 125
PIRR 126
R64CNT 642
RAMKP 1004
RCR 657
RDAYAR 654
RDAYCNT 647
REC 846
RFPR0 855
RHRAR 652
RHRCNT 645
RMINAR 651
RMINCNT 644
RMONAR 655
RMONCNT 648
RSECAR 650
RSECCNT 643
RWKAR 653
RWKCNT 646
RXPR0 854
RYRAR 656
RYRCNT 649
SAR (IIC3) 745
SCBRR 692
SCFCR 700
SCFDR 702
SCFRDR 675
SCFSR 684
SCFTDR 676
SCLSR 705
SCRSR 675
SCSCR 680
SCSMR 677
SCSPTR 703
SCTSR 676
SDBPR 1021
SDCKSCNT 234
SDCmCNT 211
SDDPWDCNT 227

SDIR 223, 1022
SDmADR 228
SDmMOD 231
SDmTR 229
SDPWDCNT 226
SDRFCNT0 220
SDRFCNT1 221
SDSTR 232
SSICR 777
SSIRDR 788
SSISR 783
SSITDR 788
STBCR 994
SYCBEEN 286
SYCBESTS1 287
SYCBESTS2 289
SYCBESW 292
SYSCR 1002
TADCOBRA_4 425
TADCOBRB_4 425
TADCORA_4 424
TADCORB_4 424
TADCR 421
TBTER 449
TBTM 419
TCBR 446
TCCR 605
TCDR 445
TCNT 425, 602
TCNTCMPCLR 406
TCNTS 444
TCORA 602
TCORB 603
TCR 380, 603
TCSR 607
TCSYSTR 431
TDDR 445
TDER 451
TEC 846
TGCR 442
TGR 426
TICCR 420
TIER 407
TIOR 387
TITCNT 448
TITCR 446
TMDR 384
TOCR 435
TOER 434
TOLBR 441
TRWER 433
TSR 412
TSTR 427
TSYR 429
TWCR 452
TXACK0 852
TXCR0 851
TXPR0 848
TXPR1 848
UMSR 857
WRCSR 630
WTCNT 627
WTCSR 628
Relationship between clock operating mode and frequency range 78
Reset state 61
Reset-synchronized PWM mode 484
Restoration from bank 156
Restoration from stack 157
Restriction on DMAC usage 727
RISC-type instruction set 26
Rounding 70
RTC crystal oscillator circuit 666

S

Saving to bank 155
Saving to stack 157
Scan mode 891
SCIF interrupt sources 725

SCIF module timing 1141
Searching cache 189
Sending a break signal 727
Serial communication interface with FIFO (SCIF) 669
Serial sound interface (SSI) 773
Setting analog input voltage 899, 908
Shift instructions 52
Sign extension of word data 26
Single mode 886
Slave receive operation 756
Slave transmit operation 753
Sleep mode 1008
Slot illegal instructions 109
Software standby mode 1009
Stack after interrupt exception handling 148
Stack status after exception handling ends 112
Standby control circuit 75
Status register (SR) 20
System control instructions 54

T

T bit 28
TAP controller 1023
Test mode settings 863
Time quanta is defined 834
Timing to clear an interrupt source 160
Transfer rate 735
Trap instructions 109
Types of exception handling and priority order 91

U

UBC trigger timing 1137
UDTDO output timing 1024
Unconditional branch instructions with no delay slot 27
User break controller (UBC) 161
User break interrupt 133
User debugging interface (H-UDI) 1019
Using interval timer mode 635
Using watchdog timer mode 634

V

Vector base register (VBR) 22

W

Watchdog timer (WDT) 625
Watchdog timer timing 1140
Write-Back Buffer (Only for Operand Cache) 192

Renesas 32-Bit RISC Microcomputer
SH7201 Group
User's Manual: Hardware

Publication Date: Rev.1.00, July 31, 2006
Rev.3.00, September 24, 2010
Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2010 Renesas Electronics Corporation. All rights reserved.
Colophon 1.0

SH7201 Group
User's Manual: Hardware

Renesas Electronics Corporation

R01UH0026EJ0300
(Previous Number: REJ09B0321-0200)

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 32-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

MB91F575BHSPMC-GSE1 MB91F594BSPMC-GSE1 PIC32MX120F032B-50I/ML MB91F464AAPMC-GSE2 MB91F577BHSPMC-GSE1 SPC5604EEF2MLH MB91F528USCPMC-GSE2 MB91F248PFV-GE1 MB91F594BPMC-GSE1 MB91243PFV-GS-136E1 MB91F577BHSPMC1-GSE1 PIC32MM0032GPL020-E/ML PIC32MM0032GPL020-E/SS MEC1632X-AUE PIC32MM0016GPL020-E/ML PIC32MM0016GPL020-E/SS PIC32MM0016GPL028-E/SS PIC32MM0016GPL028-E/SO PIC32MM0016GPL028-E/ML PIC32MM0032GPL028-E/SS PIC32MM0032GPL028-E/SO PIC32MM0032GPL028-E/ML PIC32MM0032GPL036-E/MV PIC32MM0064GPL028-E/M6 PIC32MM0016GPL028-E/M6 PIC32MM0032GPL028-E/M6 MB91F526KSEPMC-GSE1 PIC32MM0064GPL028-E/SP PIC32MM0032GPL036-E/M2 TLE9872QTW40XUMA1 FT902L-T R5F564MLCDFB#31 R5F564MLCDFC#31 R5F523E5ADFL#30 R5F524TAADFF#31 MCF51AC256ACPUE PIC32MM0064GPL028-I/ML PIC32MM0064GPL028-I/SP PIC32MM0064GPL028-I/SO PIC32MX120F032D-I/TL PIC32MX130F064D-I/ML PIC32MZ2064DAB169-I/HF PIC32MZ2064DAB288-I/4J ATUC256L4U-AUT R5F56318CDBG#U0 PIC32MX150F128C-I/TL PIC32MX170F256B-50IML PIC32MX130F064C-ITL PIC32MX230F064D-IML PIC32MM0032GPL028-I/ML